pISSN:2349-5707 Volume:12, Issue 08, August-2025 eISSN:2349-5715

COST-SENSITIVE NEURAL ARCHITECTURES FOR HANDLING CLASS IMBALANCE IN HIGH-STAKES FRAUD DETECTION SYSTEMS

Prof. Jarvinko L. Serdan
Faculty of Information Engineering, University of Maribor, Slovenia

ABSTRACT: The rapid proliferation of digital financial transactions has necessitated the development of robust automated fraud detection systems. However, these systems face a persistent challenge: the class imbalance problem, where fraudulent activities constitute a negligible fraction of total transaction volume. Traditional neural network architectures, optimized for global accuracy, frequently fail to capture these rare events, leading to financially devastating false negatives. This study investigates the efficacy of Cost-Sensitive Neural Networks (CS-NN) designed to explicitly penalize the misclassification of the minority class. By integrating a weighted cost matrix into the backpropagation error function and leveraging historical optimization techniques, we propose a framework that prioritizes high-risk sensitivity without significantly degrading overall precision. We benchmark this approach against traditional algorithms, including Support Vector Machines and standard Deep Neural Networks, utilizing datasets with varying degrees of imbalance. Our results indicate that while traditional accuracy remains comparable across models, the proposed CS-NN architecture demonstrates a statistically significant improvement in recall and F1-scores for the fraud class. Furthermore, we explore the theoretical underpinnings of dropout and regularization in the context of imbalanced learning, suggesting that standard regularization techniques must be calibrated to prevent the suppression of rare signals. The findings suggest that incorporating domain-specific cost constraints directly into the learning objective offers a more viable path for high-stakes anomaly detection than post-hoc threshold adjustment or simple data resampling.

Keywords: Fraud Detection, Class Imbalance, Cost-Sensitive Learning, Deep Neural Networks, Misclassification Cost, Backpropagation Optimization, Anomaly Detection.

1. INTRODUCTION

The digitization of global finance has introduced unprecedented efficiency to the banking sector, yet it has concurrently expanded the attack surface for malicious actors. Financial fraud, encompassing everything from credit card theft to complex money laundering schemes, represents a multi-billion dollar liability for financial institutions. As the volume of transactions scales exponentially, manual review processes have become obsolete, necessitating the deployment of automated detection systems. In recent years, Artificial Intelligence (AI) and Machine Learning (ML) have become the standard bearers for this task. However, the application of these technologies is not straightforward. As noted in recent comparative studies by Patel [1], while neural networks offer superior feature extraction capabilities compared to traditional algorithms, their application in fraud detection is often hampered by the inherent nature of the data.

The central challenge in algorithmic fraud detection is the phenomenon of class imbalance. In a typical dataset of financial transactions, legitimate activities (the negative class) may outnumber fraudulent activities (the positive class) by a factor of 10,000 to 1 or more. Chawla [17] provides a comprehensive overview of this issue, noting that standard learning algorithms are generally biased toward the majority class. This bias occurs because standard objective functions, such as Mean Squared Error (MSE) or Cross-Entropy Loss, are designed to maximize global accuracy. In a dataset where 99.9% of transactions are legitimate, a naive model can achieve 99.9% accuracy by simply classifying every transaction as legitimate. While statistically "accurate," such a model is functionally useless for fraud detection, as its recall (sensitivity) for the minority class is zero.

This problem is further compounded by the unequal costs of errors. In many machine learning tasks, such as image classification, misclassifying a cat as a dog carries the same penalty as misclassifying a dog as a cat. In financial fraud detection, however, the costs are highly asymmetrical. A False Positive (flagging a legitimate transaction as fraud) results in customer annoyance and administrative friction—a non-negligible but manageable cost. Conversely, a False Negative (failing to detect actual fraud) results in direct financial loss, potential regulatory fines, and reputational damage. Maloof [18] and Ling and Li [19] have argued that learning algorithms operating in such environments must account for these unknown or unequal costs. They suggest that the learning process itself must be modified to reflect the economic reality of the deployment environment.

The theoretical foundations for addressing these learning challenges are rooted in the early development of neural computing. The concept of adaptive elements solving difficult control problems, as explored by Barto, Sutton, and Anderson [11], laid the groundwork for systems that learn through interaction and error correction. Similarly, the work of Barlow [6] on unsupervised learning and the search for minimum entropy codes [7] suggests that neural networks can be designed to identify rare, information-rich events (like fraud) if the redundancy of the majority class is appropriately managed.

Despite these historical insights, the integration of cost-sensitive mechanisms into deep learning architectures remains an area of active research. While Zhou and Liu [20] demonstrated the feasibility of training cost-sensitive neural networks, and Khan et al. [22] extended this to deep feature representations, there remains a significant gap in understanding how these modifications interact with modern regularization techniques like Dropout, introduced by Baldi and Sadowski [2]. Furthermore, the optimization landscape of cost-sensitive loss functions is often more complex and prone to instability, requiring a re-examination of accelerated learning methods proposed by Battiti [12, 13].

This article aims to bridge this gap by proposing and evaluating a Cost-Sensitive Deep Neural Network (CS-DNN) framework. We hypothesize that by embedding the cost matrix directly into the backpropagation error derivative, we can force the network to learn features specific to the minority class without necessitating destructive data sampling techniques. We will compare this approach against standard architectures and explore the optimization dynamics that facilitate convergence in these high-stakes environments.

2. METHODS

To rigorously evaluate the proposed architecture, we employed a methodological framework that emphasizes reproducibility and strictly controlled experimental conditions. The methodology is divided into data governance, architectural design, the cost-sensitive formulation, and the optimization strategy.

2.1 Data Governance and Preprocessing

Access to real-world financial data is often restricted due to privacy regulations (GDPR, CCPA) and competitive confidentiality. Therefore, this study utilizes a high-fidelity synthetic dataset designed to mimic the statistical properties of real-world credit card transactions. The dataset was generated to reflect a severe class imbalance, with a total of 500,000 transaction records, of which only 0.17% (850 records) are labeled as fraudulent.

The feature space consists of 30 variables. Twenty-eight of these are the result of a Principal Component Analysis (PCA) transformation (V1 through V28), a common practice in financial data sharing to obscure sensitive user details while preserving the variance and correlations of the original features. The remaining two features are 'Time' (seconds elapsed since the first transaction) and 'Amount' (transaction value).

Preprocessing involved robust scaling for the 'Amount' and 'Time' features to normalize them against the PCA-transformed variables. We observed that the 'Amount' feature had a heavy right skew, typical of financial data where small transactions differ vastly from rare, high-value purchases. We applied a logarithmic transformation to reduce this skewness. The dataset was then partitioned using a stratified split to ensure the ratio of fraud to non-fraud cases remained consistent across Training (70%), Validation (15%), and Testing (15%) sets. This stratification is crucial; as noted by Chawla [17], random sampling in highly imbalanced datasets can inadvertently result in validation sets with zero minority class instances.

2.2 Foundational Architecture and Regularization

The baseline model for our experiment is a fully connected Deep Neural Network (DNN). The architecture consists of an input layer matching the dimensionality of the feature space (30 neurons), followed by four hidden layers with 64, 32, 16, and 8 neurons respectively. The reduction in layer size creates a "bottleneck" structure, encouraging the network to learn compressed, high-level representations of the transaction data, echoing the principles of minimum entropy coding described by Barlow et al. [7].
The activation function for the hidden layers is the Rectified Linear Unit (ReLU), chosen for its ability to mitigate the vanishing gradient problem. The output layer utilizes a Sigmoid activation function to output a probability score between 0 and 1.
A critical component of our architecture is the implementation of Dropout, as formalized by Baldi and Sadowski [2]. In imbalanced learning, neural networks are highly prone to overfitting the majority class. Without intervention, the network effectively "memorizes" the characteristics of legitimate transactions. We introduced Dropout layers with a rate of 0.5 after each hidden layer. Baldi and Sadowski demonstrated that Dropout can be interpreted as an implicit ensemble method, training a multitude of thinned networks simultaneously. In our context, this prevents the network from relying too heavily on any single feature that might be correlated with the majority class, thereby forcing it to seek more robust features that discriminate the minority class.

2.3 Cost-Sensitive Learning Framework

The core innovation in this study is the modification of the learning objective. Standard neural network training minimizes a loss function, typically Binary Cross-Entropy (BCE).
In a cost-sensitive framework, we modify this loss function to weigh the errors differently. We define a cost matrix C , where C_FN is the cost of a false negative and C_FP is the cost of a false positive. Following the recommendations of Elkan and the applications discussed by Zhou and Liu [20], we assign a significantly higher weight to the minority class.
This weighting scheme is integrated directly into the backpropagation algorithm. When the network computes the gradient of the loss with respect to the weights, the error term for the positive class (fraud) is multiplied by the cost factor. Consequently, a failure to detect a fraud instance results in a gradient update that is orders of magnitude larger than a failure to correctly classify a legitimate transaction. This forces the optimization trajectory to prioritize the reduction of fraud-related errors.

2.4 Optimization and Hyperparameter Tuning

The optimization of cost-sensitive neural networks presents unique challenges regarding convergence stability. Because the gradients associated with minority class errors are artificially amplified, the optimization landscape becomes rugged, with steep cliffs that can cause the parameters to oscillate or diverge if the learning rate is not carefully managed. This necessitates a sophisticated approach to gradient descent, drawing on the historical analysis of second-order methods by Battiti [13] and accelerated learning schemes [12].
Standard Stochastic Gradient Descent (SGD) often struggles in these environments because a single batch containing a high-weight fraud example can drastically shift the weights, undoing previous learning on the majority class. To mitigate this, we employed the Adam optimizer, which adapts the learning rates for each parameter based on the first and second moments of the gradients. However, simply using Adam is insufficient without considering the batch composition.
We observed that the stability of the learning process is heavily dependent on the "effective batch size" of the minority class. If a mini-batch contains zero fraud examples, the cost-sensitive weights are inactive, and the network learns in standard mode. If the next batch contains several fraud examples, the weighted loss spikes. This variance introduces noise into the gradient estimation. To address this, we implemented a dynamic learning rate scheduler that decays the learning rate when the validation loss plateaus.
Furthermore, we revisited the concepts of modular learning proposed by Ballard [3]. While we did not implement distinct modules for different transaction types, the layer-wise architecture was essentially treated as a hierarchy of feature extractors. The lower layers learn basic statistical correlations, while the

upper layers, closer to the output, make the decision based on the cost-weighted risk.
The choice of the cost ratio is a hyperparameter itself. We experimented with cost ratios ranging from 1:1 (standard learning) to 100:1. The selection of these ratios was guided by the theoretical work of Lawrence et al. [21], who discussed the relationship between prior class probabilities and network outputs. They suggest that the network's output probability is implicitly biased by the training set priors. By introducing cost weights, we are effectively shifting the decision boundary to compensate for this prior bias.

2.5 Theoretical Expansion: Optimization Stability in Non-Convex Landscapes

To fully understand the mechanics of our proposed Cost-Sensitive DNN, one must delve deeper into the optimization difficulties inherent in this approach. The error surface of a neural network is non-convex, containing multiple local minima. In a balanced dataset, these minima are generally distributed in a way that allows standard optimizers to find a solution that generalizes well. However, in a highly imbalanced, cost-weighted scenario, the error surface is distorted. The "valleys" corresponding to solutions that correctly classify the minority class become extremely narrow and deep due to the high penalty weights.
Battiti [12] discussed accelerated backpropagation methods that attempt to navigate these surfaces by using information about the curvature of the error function (second-order information). While full Newton-method optimization is computationally prohibitive for modern deep networks due to the cost of computing the Hessian matrix, the insights regarding curvature are relevant. When we apply a cost weight of 100 to a fraud sample, we are effectively increasing the curvature of the error surface in the direction of the weights responsible for that classification.
This high curvature means that a standard step size (learning rate) might overshoot the minimum. This phenomenon explains why cost-sensitive networks often exhibit "exploding gradients" or erratic loss curves. To counteract this, we utilized gradient clipping, a technique where the norm of the gradient vector is capped at a threshold before the weight update is applied. This prevents the massive error signals from the minority class from destabilizing the entire network structure.
Additionally, we must consider the implications of Baxter and Bartlett's work on infinite-horizon policy-gradient estimation [16]. While their work focuses on reinforcement learning, the parallel is strong. In our case, the "policy" is the classification decision, and the "reward" is the negative cost. The high variance in the gradient estimates (caused by the rarity of the fraud signal) is analogous to the variance in policy gradient methods. This similarity suggests that techniques used in RL, such as baseline subtraction or advantage estimation, could theoretically be adapted for supervised imbalanced learning to reduce variance, although such adaptation is beyond the scope of the current experimental setup.
We also draw upon the concept of "Learning Receptive Fields" as described by Barrow [8]. In the early layers of our network, the neurons are learning to respond to specific combinations of PCA features. In a standard network, the receptive fields would evolve to capture features maximizing variance in the majority class. In our cost-sensitive network, the backpropagated error from the minority class is strong enough to rotate these receptive fields towards features that are distinctive of fraud, even if those features are statistically rare in the overall dataset. This alignment is critical and validates the hypothesis that cost-sensitive learning alters the internal representation of the data, not just the final decision threshold.

3. RESULTS

The experimental results provide a quantitative validation of the Cost-Sensitive DNN (CS-DNN) approach compared to the baseline models. All models were trained for 50 epochs with early stopping criteria enabled to prevent overfitting.

3.1 Performance Metrics Evaluation

We avoided using accuracy as a primary metric due to the reasons outlined in the introduction. Instead, we focused on Precision, Recall, the F1-Score (the harmonic mean of precision and recall), and the Area Under the Precision-Recall Curve (AUPRC).
The baseline Standard DNN (with equal costs) achieved an accuracy of 99.89%. However, its recall for the fraud class was only 0.58. This indicates that while the model rarely made mistakes on legitimate

transactions, it missed 42% of actual fraud cases. This performance is unacceptable in a high-stakes financial context.

The Support Vector Machine (SVM) and Random Forest models performed marginally better on precision but struggled with recall. The SVM, in particular, showed a tendency to be overly conservative, likely due to the difficulty of establishing a separating hyperplane when the minority class is so sparse in the high-dimensional space.

The proposed CS-DNN, trained with a cost ratio of 10:1 (Fraud:Legitimate), showed a marked improvement. The recall for the fraud class increased to 0.82, while maintaining a precision of 0.78. When the cost ratio was increased to 50:1, the recall further improved to 0.91. However, this came at a trade-off: the precision dropped to 0.65. This inverse relationship is expected; as the model becomes more aggressive in hunting fraud, it flags more borderline legitimate transactions as suspicious.

3.2 Impact of Cost Matrices

We conducted a sensitivity analysis by varying the cost matrix. We observed a non-linear relationship between the cost weight and the recall improvement. Increasing the weight from 1:1 to 10:1 yielded the most significant marginal gain. Beyond 50:1, the gains in recall diminished, while the number of False Positives exploded. This suggests a point of diminishing returns where the network begins to over-prioritize the minority class to the detriment of the overall system utility.

It is interesting to note the findings of Lawrence et al. [21] regarding prior probabilities. Our results confirm that the optimal cost ratio is roughly inversely proportional to the class imbalance ratio, but requires fine-tuning. Simply setting the cost ratio equal to the imbalance ratio (e.g., 500:1) destabilized the training, causing the model to predict "Fraud" for almost everything, which validates the need for the gradient clipping and stabilization techniques discussed in the Methods section.

3.3 Convergence Analysis

In analyzing the training curves, the Standard DNN converged smoothly and quickly. The CS-DNN exhibited more volatility in the validation loss during the early epochs. This volatility is a direct result of the high-variance gradient updates discussed earlier. However, with the Adam optimizer and learning rate decay, the CS-DNN eventually settled into a stable minimum.

Comparing our results to the recent work of Patel [1], we found that while Patel's comparison favored neural networks over traditional algorithms generally, our specific implementation of cost-sensitive neural networks outperformed the generic neural network baselines used in his study. This reinforces the argument that architecture alone is insufficient; the loss function must be tailored to the problem domain.

4. DISCUSSION

The results of this study illuminate the complex dynamics of learning from imbalanced data and highlight the efficacy of cost-sensitive mechanisms in deep neural networks.

4.1 Interpretation of Cost-Sensitivity

The primary finding is that embedding costs into the learning objective is a more effective strategy than simple threshold adjustment. When a standard network is trained and the decision threshold is lowered post-hoc to increase recall, the network is essentially being forced to operate in a region of its decision space where it has low confidence. In contrast, the CS-DNN learns a decision boundary that is structurally different. By penalizing missed fraud during training, the network learns feature representations that are inherently more discriminative of fraud.

This aligns with the work of Zhou and Liu [20], who argued that cost-sensitive learning modifies the feature space itself. Our analysis of the activations in the hidden layers suggests that the CS-DNN dedicates more neurons to detecting "outlier" features than the standard DNN, which dedicates most of its capacity to modeling the manifold of legitimate transactions.

4.2 The Stability-Plasticity Dilemma and Concept Drift

A major challenge in fraud detection that extends beyond the static dataset analysis is the temporal nature of fraud. Fraud patterns change rapidly as attackers adapt to defense mechanisms. This touches upon the "stability-plasticity dilemma" in neural systems—how to learn new patterns (plasticity) without forgetting old ones (stability).

While our experiment used a static dataset, the theoretical works of Barto and Mahadevan [9] on hierarchical learning and Baluja [4] on population-based incremental learning offer insights into how this system could be extended. A static CS-DNN might become obsolete as fraud tactics evolve (concept drift). A potential solution lies in modular or incremental learning, where the cost matrix is dynamically adjusted, or where new network modules are added to learn novel fraud types without disrupting the weights associated with established fraud patterns.

The issue of concept drift is particularly insidious in cost-sensitive learning. If the characteristics of the minority class change, the highly weighted neurons tracking the "old" fraud patterns may produce high confidence False Positives on new, benign behaviors that resemble the old fraud. Conversely, new fraud patterns might not trigger the specific receptive fields learned during the initial high-cost training. This suggests that continuous retraining is not just a maintenance task but a fundamental requirement of the system. The work by Baum and Petrie [15] on statistical inference for probabilistic functions provides a mathematical basis for detecting when the underlying distribution has shifted sufficiently to warrant a model update.

4.3 Feature Engineering and Deep Representation

An important aspect of our discussion must center on the role of feature engineering versus deep representation learning. In traditional fraud detection (as referenced in Ling and Li [19]), heavy emphasis was placed on manual feature engineering—creating variables like "velocity of transactions" or "geographic distance." The promise of Deep Learning (Khan et al. [22]) is that the network should learn these features automatically.

Our results support this to an extent, but with a caveat. The CS-DNN was able to extract useful signals from the PCA-transformed variables, which are abstract mathematical representations. This indicates that the network is finding non-linear correlations that would be difficult for a human analyst to define manually. However, the inclusion of explicit "Time" and "Amount" features remained critical. This suggests a hybrid approach is likely optimal: providing the network with known, high-value engineered features while allowing it to discover latent features within the high-dimensional data.

The concept of "minimum entropy codes" by Barlow [7] is relevant here. Fraudulent transactions can be viewed as high-entropy events within a low-entropy stream of legitimate behavior. The CS-DNN essentially learns a coding scheme where these high-entropy events result in a distinct activation pattern. The cost function forces the network to assign a "code" to fraud that is maximally distinct from the code for legitimate transactions, even if that requires distorting the representation space.

4.4 Operational and Ethical Implications

Implementing a CS-DNN in a real-world production environment requires careful consideration of operational constraints. The increase in False Positives observed at high cost ratios represents a real operational cost. If a bank flags too many legitimate transactions, customer trust erodes. Therefore, the "optimal" model is not necessarily the one with the highest F1-score, but the one that maximizes profit (fraud saved minus administrative costs and customer churn).

This leads to an ethical consideration regarding the "black box" nature of these models. As Balzer [5] noted in his perspective on automatic programming, as systems become more complex, understanding their internal logic becomes harder. If a CS-DNN denies a transaction based on a complex, non-linear interaction of features, the institution must be able to explain why, especially under regulations like the Equal Credit Opportunity Act. The high weighting of minority class features might inadvertently lead the model to proxy protected characteristics (like age or location) as fraud indicators if they are correlated in the training data.

4.5 Limitations and Future Directions

The primary limitation of this study is the use of synthetic data. While statistically representative, it lacks the nuance and "dirty" nature of real-world logs (e.g., labeling errors). Future research should focus on "Learning with Noisy Labels" in a cost-sensitive framework.

Furthermore, the exploration of Reinforcement Learning (RL) offers a promising avenue. Barto, Singh, and Chentanez [10] discussed intrinsically motivated learning. An RL agent could be designed to actively "hunt" for fraud, treating the investigation process as a sequential decision problem. Instead of a static classification, the system could request more authentication steps (step-up authentication) for ambiguous transactions. This moves the problem from simple classification to sequential decision making under uncertainty.

Finally, the integration of genetic search based optimization, as proposed by Baluja [4], could be used to optimize the cost matrix itself. Rather than manually tuning the costs, an evolutionary algorithm could evolve the cost matrix to maximize a holistic business metric.

5. CONCLUSION

This study has demonstrated that Cost-Sensitive Deep Neural Networks represent a significant advancement over traditional methods for fraud detection in imbalanced datasets. By embedding the economic reality of the problem directly into the mathematical objective of the network, we can achieve superior sensitivity to rare, high-impact events. While challenges regarding optimization stability and false positive rates remain, the integration of robust regularization and adaptive optimization techniques provides a viable path forward. As financial systems continue to automate, such "risk-aware" algorithms will be essential in maintaining the integrity of the global transaction infrastructure.

REFERENCES

1. Dip Bharatbhai Patel. (2025). Comparing Neural Networks and Traditional Algorithms in Fraud Detection. The American Journal of Applied Sciences, 7(07), 128–132. https://doi.org/10.37547/tajas/Volume07Issue07-13
2. Baldi, P., & Sadowski, P. (2014). The dropout learning algorithm. Artificial Intelligence, 210C, 78–122.
3. Ballard, D. H. (1987). Modular learning in neural networks. In Proc. AAAI (pp. 279–284).
4. Baluja, S. (1994). Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report CMU-CS-94-163. Carnegie Mellon University.
5. Balzer, R. (1985). A 15 year perspective on automatic programming. IEEE Transactions on Software Engineering, 11(11), 1257–1268.
6. Barlow, H. B. (1989). Unsupervised learning. Neural Computation, 1(3), 295–311.
7. Barlow, H. B., Kaushal, T. P., & Mitchison, G. J. (1989). Finding minimum entropy codes. Neural Computation, 1(3), 412–423.
8. Barrow, H. G. (1987). Learning receptive fields. In Proceedings of the IEEE 1st annual conference on neural networks, vol. IV (pp. 115–121). IEEE.
9. Barto, A. G., & Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. Discrete Event Dynamic Systems, 13(4), 341–379.
10. Barto, A. G., Singh, S., & Chentanez, N. (2004). Intrinsically motivated learning of hierarchical collections of skills. In Proceedings of international conference on developmental learning (pp. 112–119). Cambridge, MA: MIT Press.
11. Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. IEEE Transactions on Systems, Man and Cybernetics, SMC-13, 834–846.
12. Battiti, R. (1989). Accelerated backpropagation learning: two optimization methods. Complex Systems, 3(4), 331–342.
13. Battiti, T. (1992). First- and second-order methods for learning: between steepest descent and Newton’s

method. Neural Computation, 4(2), 141–166.
14. Baum, E. B., & Haussler, D. (1989). What size net gives valid generalization? Neural Computation, 1(1), 151–160.
15. Baum, L. E., & Petrie, T. (1966). Statistical inference for probabilistic functions of finite state Markov chains. The Annals of Mathematical Statistics, 1554–1563.
16. Baxter, J., & Bartlett, P. L.(2001). Infinite-horizon policy-gradient estimation. Journal of Artificial Intelligence Research, 15(1), 319–350.
17. Nitesh V Chawla. Data mining for imbalanced datasets: An overview. In Data mining and knowledge discovery handbook, pages 853–867. Springer, 2005.
18. Marcus A Maloof. Learning when data sets are imbalanced and when costs are unequal and unknown. In ICML-2003 workshop on learning from imbalanced data sets II, volume 2, pages 2–1, 2003.
19. Charles X Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In KDD, volume 98, pages 73–79, 1998.
20. Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering, 18(1):63–77, 2006.
21. Steve Lawrence, Ian Burns, Andrew Back, Ah Chung Tsoi, and C Lee Giles. Neural network classification and prior class probabilities. In Neural networks: tricks of the trade, pages 299–313. Springer, 1998.
22. Salman H Khan, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri. Cost sensitive learning of deep feature representations from imbalanced data. arXiv preprint arXiv:1508.03422, 2015