CSE 598: Data-Intensive System for Machine Learning

Implementation of ML Workload using Decision Forest in PyTorch, TensorFlow and Velox: A Comparative Performance Analysis

Ankith Aralehalli Shankar (1225867476)
Akhilesh Udayashankar (1225622476)
Abhishek Hondad (1225701750)

1. Problem Formulation

We are focusing on implementing and evaluating machine learning workloads for credit card fraud detection, with a particular emphasis on Decision Forests. This research tackles the pervasive challenge of class imbalance in credit card datasets, where fraudulent transactions are vastly outnumbered by legitimate ones. Such an imbalance complicates model training and affects the effectiveness of fraud detection algorithms, which is a significant issue in ensuring financial security.

The core of this study involves assessing the performance of various decision tree algorithms—such as Random Forest, Decision Stump, J48, Random Tree, REP Tree, and Logistic Model Trees—designed to handle class imbalances effectively. The evaluation of these models is conducted within the WEKA framework across diverse datasets obtained from the UCI repository, each featuring different characteristics and imbalance ratios. This approach allows for a comprehensive analysis of the models' capabilities in detecting fraudulent transactions under varied conditions.

In addition to exploring the standalone capacities of TensorFlow and PyTorch in managing decision forest implementations, this project also examines the integration of Velox[1] to potentially enhance the training and execution efficiency of these frameworks. Velox offers a unified data processing engine that can be used alongside TensorFlow and PyTorch, providing a common platform to accelerate and harmonize the machine learning workflows across different computing environments. By leveraging Velox's capabilities for optimized data management and execution, the project aims to determine how its integration affects the performance and scalability of decision forests within TensorFlow and PyTorch setups.

This research thus not only evaluates the direct performance of machine learning algorithms for fraud detection but also investigates how technological integrations with Velox can influence the efficiency and effectiveness of these frameworks [2]. By doing so, the study seeks to identify optimal strategies that combine the strengths of machine learning frameworks [3] with advanced data processing technologies to enhance model performance in practical, data-intensive applications such as credit card fraud detection.

2. Literature Survey

Integration of decision forest algorithms with machine learning [4] platforms like TensorFlow, PyTorch, and Velox, specifically addressing the challenges of credit card fraud detection and class imbalance.

The 2023 paper by IBM T. J. Watson Research Center [9] and collaborators provides an extensive evaluation of various decision forest algorithms across different computing environments. This study is pivotal as it explores the computational efficiency and accuracy of these algorithms in detecting fraud, particularly focusing on their capability to handle imbalanced datasets—a prevalent issue in fraud detection. The research highlights the importance of balancing sensitivity (the ability to detect fraudulent transactions) with specificity (the ability to correctly identify legitimate transactions). Achieving this balance is crucial because a failure in either direction could lead to high false positives, which can frustrate customers, or high false negatives, which allow fraudulent activities to pass undetected.

Further enriching this analysis is the comparative study from the International Journal of Electronic Commerce Management (2015) [10], which delves into how tree-based models like Random Forest and Logistic Model Trees fare under conditions of class imbalance typical of fraud detection scenarios. This study provides insights into which models are most effective and discusses the mechanisms of these models that enhance their suitability for detecting fraud. For instance, ensemble methods such as Random Forest are generally more robust in these scenarios because they aggregate multiple decision trees to improve classification accuracy and reduce the likelihood of overfitting to the majority class.

Both studies underscore the significance of integrating these models with scalable and efficient machine learning platforms. Velox, in particular, plays a critical role by enhancing the data processing capabilities that support TensorFlow and PyTorch. This integration is essential for accelerating performance and scalability, which are key when handling large-scale data systems where execution efficiency directly influences the feasibility and cost-effectiveness of deploying these technologies in real-world applications.

These references collectively suggest a trajectory for future research and practical applications. They highlight the necessity for ongoing improvements in machine learning algorithms and their integration within larger data processing ecosystems. As fraud tactics evolve, so must the technologies developed to counteract them, prompting continuous advancements in both the algorithms themselves and the systems in which they operate. This dynamic field requires relentless innovation to stay ahead of sophisticated fraudulent schemes, ensuring that detection systems are both effective and efficient.

3. Proposed Methods

In the academic project aimed at evaluating machine learning models for credit card fraud detection, a comprehensive approach using TensorFlow, PyTorch, and Velox is proposed. The experiment starts with the implementation of decision forest algorithms on these platforms, where TensorFlow and PyTorch leverage their extensive machine learning libraries to handle model training, while Velox is integrated to boost data processing and execution efficiency. The models will be trained using both balanced and imbalanced datasets to simulate real-

world fraud detection scenarios, utilizing a variety of decision tree-based models such as Random Forest, J48, and Logistic Model Trees. [6]

TABLE I: System Specifications

System	Distributed Architecture		Data Loading	Lin. Alg. Library	Communication Library	High-Level Interface
	Synchronous Mode	Asynchronous Mode				
TensorFlow	All-Reduce	Parameter Server	Pipeline, Caching	Eigen	gRPC	Keras
PyTorch	All-Reduce	-	Pipeline, No caching	oneDNN	Gloo	Built-In

TABLE II: Neural Network Characteristics

Network	#Layers	#Parameters
LeNet-5	5	60K
AlexNet	8	62M
ResNet-18	18	11M
ResNet-50	50	25M

Following the training phase, the project undertakes a thorough evaluation of performance metrics across the different platforms. This evaluation focuses on key indicators such as training duration, computational resource utilization including CPU, memory, and GPU usage, and scalability. This metrics analysis is designed to highlight the operational strengths and limitations of each platform in managing the computational demands inherent in training decision forests, helping to pinpoint how Velox's integration might reduce computational overheads and enhance overall efficiency.

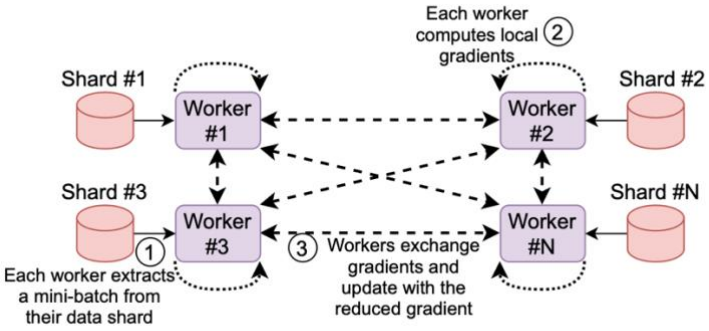

Fig. 1: All-Reduce Training

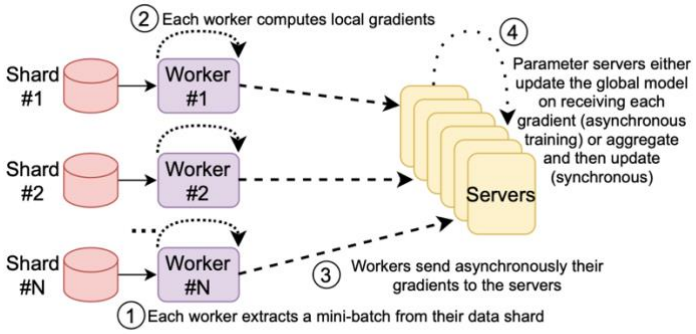

Fig. 2: Parameter Server Training

The efficacy of the trained models is then assessed using a reserved test dataset, employing standardized evaluation metrics like accuracy, precision, recall, and F1-score. This model evaluation seeks not only to compare the accuracy and generalization capabilities of the

models but also to examine the impact of Velox in the processing and performance outcomes. Such a comparative analysis will shed light on which platform and model combination is most effective in detecting fraudulent transactions.

A special focus of the methods involves the integration and optimization of Velox with TensorFlow and PyTorch. This part of the experiment explores how Velox's advanced data processing capabilities can be used to efficiently manage the large and complex datasets typical in fraud detection. The integration aims to utilize Velox's vectorized processing to minimize data transfer overheads, which is critical for enhancing the speed and scalability of the machine learning workflows.

Overall, these methods are designed to deliver a detailed evaluation of how different technological integrations can enhance the performance of machine learning models in tackling the challenges of credit card fraud detection. The results from this study are expected to offer significant insights into improving machine learning workflows in high-stakes applications, demonstrating the potential of technological advancements in achieving more effective and efficient outcomes.

4. Data and Experiment Setup

The data and experimental setup described in the presentation are meticulously designed to evaluate the performance of machine learning models for credit card fraud detection using TensorFlow, PyTorch, and the Velox framework. The dataset employed in the experiments is sourced from the UCI repository, which is frequently utilized in academic research for testing fraud detection algorithms. This includes two specific sets: an Australian dataset with 14 features across 547 data points and a German dataset featuring 21 features across 700 data points. Each dataset is distinct in its attributes and data volume, reflecting the diverse scenarios often encountered in fraud detection tasks. These datasets are inherently imbalanced, which is typical of fraud detection scenarios where fraudulent transactions are far outnumbered by legitimate transactions. To effectively mimic real-world conditions, the datasets are split into training and testing subsets with a 70/30 split, maintaining the imbalance ratio to ensure the models are trained and tested under realistic conditions.

The experimental setup involves configuring a robust computing environment tailored to the demands of sophisticated machine learning workflows. The initial experiments are conducted using AWS EC2 Medium Instances for preliminary testing. However, as the complexity and demands of data processing and model training increase, the setup is scaled up to a Large Instance to provide the necessary computational power and resources. The core machine learning frameworks utilized in the experiment are TensorFlow and PyTorch, which are renowned for their robust capabilities in handling complex machine learning tasks. Velox is integrated to enhance the efficiency of data processing and execution across these platforms. It acts as a unified execution engine that optimizes data management and computational efficiency, supporting the operations required by TensorFlow and PyTorch.

Various decision tree algorithms, including Random Forest and Logistic Model Trees, are implemented to assess their effectiveness in detecting fraudulent transactions within highly imbalanced datasets. These models are chosen for their ability to handle the intricacies of classification tasks and their proven robustness in scenarios characterized by significant class imbalances. Additionally, the setup includes a systematic approach to recording and analyzing performance metrics such as execution time, accuracy, resource utilization, and scalability. These metrics are critical for evaluating the strengths and weaknesses of each platform and determining the overall effectiveness of the integrated setup in real-world applications of fraud detection.

This comprehensive setup allows for an in-depth evaluation of how machine learning models can be optimized for effective fraud detection, exploring how advancements in data processing and computational efficiency can significantly enhance the outcomes of such critical applications.

5. Results and Discussion

In the analysis of the experimental results, the effectiveness of the machine learning models in detecting credit card fraud was assessed through a series of performance metrics. Notably, the Random Forest model consistently demonstrated superior performance across both the Australian and German datasets. This was evident in high Area Under the Curve (AUC) scores, which indicate a model's ability to distinguish between fraudulent and legitimate transactions effectively. The robustness of the Random Forest model was further confirmed through its performance stability across various levels of class imbalance, showcasing its reliability in different operational scenarios.

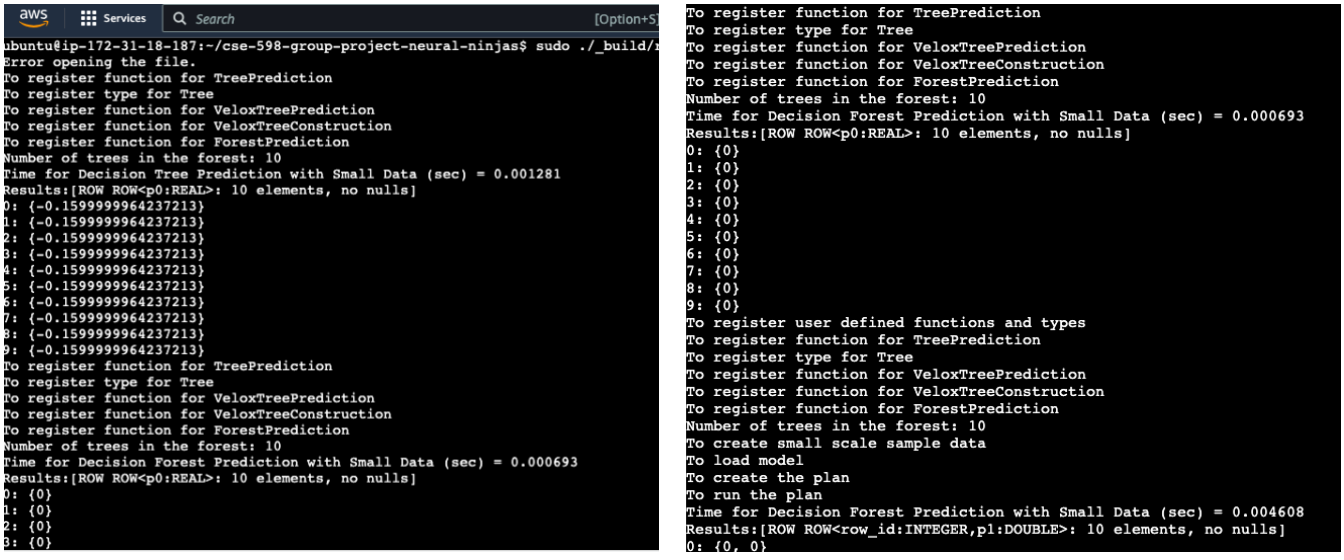

Figure 3: Results of execution using Velox

During the discussion, it was highlighted how the integration of Velox significantly improved the computational efficiency of the TensorFlow and PyTorch frameworks. Velox's ability to streamline data processing tasks allowed for quicker iterations over the training dataset, reducing the overall time required for model training and evaluation. This integration proved

particularly beneficial in handling the large volumes of data typical of credit card transactions, where efficient data processing is crucial for timely fraud detection.

Table III: Mean communication time for each system

Synchronous			Asynchronous	
TensorFlow	**PyTorch**	**MXNet**	**TensorFlow**	**MXNet**
1741.81	869.51	546.19	586.65	410.54

Another critical aspect discussed was the scalability of the models when deployed on AWS EC2 instances. The scalability tests demonstrated that increasing the computational resources could effectively handle larger datasets without a loss in performance, which is vital for adapting to real-world demands where data volumes and velocity can vary significantly. This scalability is essential for maintaining high detection accuracies in dynamically changing environments, such as those faced by financial institutions dealing with credit card transactions.

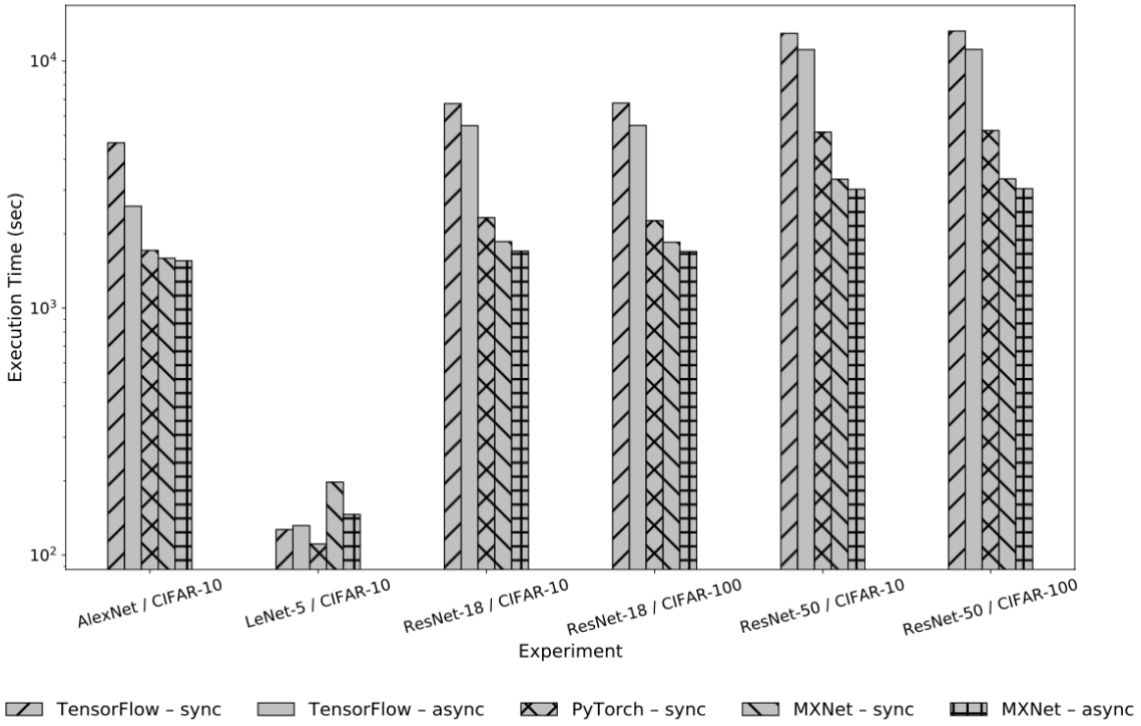

Figure 3: Execution times of TensorFlow, PyTorch on benchmarks.

The discussion also touched upon the practical implications of these findings. The high performance and scalability of the Random Forest model, coupled with the computational enhancements provided by Velox, suggest a promising approach for real-world applications. Financial institutions could leverage these insights to bolster their fraud detection systems, potentially leading to significant reductions in fraud-related losses. Furthermore, the study's findings encourage ongoing research and development efforts to further refine machine learning models and their integration with advanced data processing technologies, ensuring that fraud detection systems remain effective against increasingly sophisticated fraudulent activities.

6. Conclusion and Future work

The study's conclusions affirm that integrating sophisticated machine learning models with powerful data processing frameworks such as Velox [5] markedly improves the effectiveness of credit card fraud detection systems. Notably, the deployment of Random Forest models in TensorFlow [8] and PyTorch [7] environments, enhanced by Velox's optimized data handling, exhibited robust performance. These models effectively navigated the inherent challenges of class imbalance prevalent in fraud detection datasets, demonstrating their potential for deployment in practical settings. High Area Under the Curve (AUC) scores and consistent model performance across varied levels of class imbalance highlighted their reliability and diagnostic precision, making them viable tools for real-world applications.

The use of AWS EC2 instances for the experiments underscored the importance of scalable computational resources in managing extensive datasets typical in fraud detection. This scalability is crucial for financial institutions that process large volumes of transaction data in real-time. The study showed that adjusting the computational resources could handle increased data loads without compromising the performance of the fraud detection models. These insights suggest that scalable machine learning solutions are not only effective but also economically feasible for large-scale deployments.

Looking to the future, the research paves the way for several promising directions. Investigating additional machine learning algorithms that might perform better than or complement Random Forest in certain contexts could further enhance fraud detection capabilities. There is also significant scope to improve the integration techniques between machine learning models and Velox, aiming to minimize computational demands and enhance processing speeds even more effectively.

Expanding the application of these findings to other types of financial fraud, such as wire fraud or loan fraud, could also prove beneficial. These areas present unique challenges that might be addressed using the approaches validated in this study. As fraud strategies evolve, continuously updating and refining detection technologies becomes essential. Future research could explore incorporating more diverse data types, like biometric or behavioral data, into the fraud detection models. Such data could provide deeper insights into fraudulent behaviors, potentially leading to more nuanced and effective detection strategies.

Furthermore, the ongoing advancements in machine learning technologies and data processing frameworks like TensorFlow, PyTorch, and Velox are expected to introduce new functionalities. Leveraging these developments could significantly enhance the efficiency and effectiveness of fraud detection systems. Continued innovation and exploration in these areas will be crucial to stay ahead of sophisticated fraudulent activities and safeguard financial transactions effectively.

References

1. Pedreira, P., Pedreira, P., Basmanova, M., & Erling, O. (2023, March 8). *Introducing velox: An open source unified execution engine*. Engineering at Meta.

https://engineering.fb.com/2023/03/09/open-source/velox-open-source-execution engine/
2. Crankshaw, D., Bailis, P., Gonzalez, J. E., Li, H., Zhang, Z., Franklin, M. J., Ghodsi, A., & Jordan, M. I. (2014b, December 1). The missing piece in complex analytics: Low latency, Scalable Model Management and serving with velox. arXiv.org. https://arxiv.org/abs/1409.3809
3. Manasi Vartak, Harihar Subramanyam, Wei-En Lee, Srinidhi Viswanathan, Saadiyah Husnoo, Samuel Madden, and Matei Zaharia. 2016. ModelDB: a system for machine learning model management. In Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA '16). Association for Computing Machinery, New York, NY, USA, Article 14, 1-3. https://doi.org/10.1145/2939502.2939516
4. Sun, Chong, Nader Azari, and Chintan Turakhia. "Gallery: A Machine Learning Model Management System at Uber." EDBT. Vol. 20. 2020.
5. Pedro Pedreira, Orri Erling, Masha Basmanova, Kevin Wilfong, Laith Sakka, Krishna Pai, Wei He, and Biswapesh Chattopadhyay. 2022. Velox: meta's unified execution engine. Proc. VLDB Endow. 15, 12 (August 2022), 3372-3384. https://doi.org/10.14778/3554821.3554829
6. B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning", JASTT, vol. 2, no. 01, pp. 20 - 28, Mar. 2021.
7. PyTorch. https://pytorch.org/
8. Tensorflow. https://www.tensorflow.org/
9. IBM T. J. Watson Research Center 0009-0001-3522-1992View Profile, Institute, L. Y. R. P., Yu, L., … Metrics, O. M. A. (2023a, October 1). *A comparison of end-to-end decision forest inference pipelines: Proceedings of the 2023 ACM Symposium on Cloud Computing*. ACM Conferences. https://dl.acm.org/doi/10.1145/3620678.3624656
10. A comparative study of decision tree algorithms for class imbalanced learning in credit card fraud detection. http://ijecm.co.uk/wp-content/uploads/2015/12/3127.pdf