Using Embeddings to Train and Build Privacy-Preserving Machine Learning Models and Applications

1st Pierce Buckner-Wolfson
School of Engineering
Wesleyan University
Middletown, CT, USA
ORCID: 0009-0008-3113-2133

2nd Dalmo Cirne
Machine Learning Organization
Workday, Inc.
Boulder, CO, USA
ORCID: 0009-0005-6354-0041

***Abstract*—Protecting data privacy is a critical responsibility for application developers. However, without data, it becomes impossible to build entire categories of products. This paper proposes an innovative method for training Machine Learning (ML) models using only embeddings (derived data). Embeddings represent the original data as multidimensional vectors and, as such, can be plotted and clustered in hyperspace, enabling effective solutions for problems such as anomaly detection, search, sentiment analysis, recommendation, and graph prediction, without requiring access to the raw data. Using only derived representations of the data and their clustering patterns, this approach preserves privacy while allowing responsible application development. Many machine learning algorithms such as Neural Networks, Gradient Boosting, and K-Nearest Neighbors are well-suited tools, providing cost-effective and computationally efficient alternatives to Large Language Models (LLMs). The approach proposed here balances data-driven innovation that is compliant with strict privacy requirements while unlocking the space for the development of powerful applications.**

***Index Terms*—Embeddings, Machine Learning, Privacy.**

I. Introduction

A central challenge in modern machine learning is balancing the field's dependence on vast datasets with the critical need for safeguarding user privacy. As ML models become integral to sensitive domains like finance and healthcare, the need for tools to protect this data has led to the emergence of Privacy-Preserving Machine Learning (PPML) [1].

A commonly used privacy-preserving technique is to analyze data at an aggregate level, such as looking at distributions or summary statistics. Although this approach is inherently safe, it sacrifices the fine-grained individual-level detail required to build robust predictive systems.

With these concerns in mind, this paper proposes the use of embeddings as a practical and effective solution. Embedding sensitive data acts as a proxy layer, transforming it into high-dimensional vectors that are not suitable for human interpretation. Moreover, these vectors preserve the original relational patterns in the raw data, making them highly useful for computation. By operating on these vector representations, ML models can learn from the underlying structure of the data to perform tasks like *anomaly detection*, by identifying embeddings outside typical clusters, *recommendation*, by suggesting the typical values in the cluster, or *classification*, by observing patterns. All without accessing the raw sensitive inputs.

These are some examples of how embeddings can be used to provide solutions to a broad range of use cases:

- **Anomaly detection**
  - General purpose anomaly detection engine (i.e., risk score calculator).
  - Text, tags, and non-continuous data are clustered by semantic similarity.
  - Simple fraud detection.
- **Recommendations**
  - Similar and frequently used items are suggested during data entry.
  - Resource management (assets and resource forecasting).
- **Classification**
  - Extract keywords from text and use them to categorize content (e.g., procurement, expenses, accounts payable, categorize memos).

These embedding patterns can be effectively leveraged by a number of efficient machine learning models, including Gradient Boosting [2], Neural Networks [3], and K-Nearest Neighbors [4]. Although an LLM may solve similar problems when augmented with techniques such as Retrieval Augmented Generation (RAG) [5] and Low-Rank Adaptation (LoRA) [6], the models proposed here offer a more scalable and computationally efficient pathway for many real-world applications.

This paper empirically evaluates the embedding-based approach on a real-world financial task: loan default prediction. The results demonstrate that this method is highly competitive with a baseline model trained on raw data, validating its use as a foundational step for building effective and more responsible PPML systems.

This paper is organized as follows. Section II details the system architecture, data pipeline, embeddings pipeline, and model framework. Section III presents experimental results on

loan default prediction using XGBoost and Neural Networks. And Section IV concludes with key findings and future work.

II. Methodology

A. System Architecture

The architecture diagram for the proposed privacy-preserving machine learning system is shown in Figure 1. This blueprint serves as the foundation for the methodology detailed in this paper. The workflow, designed to prevent data exposure, proceeds in several distinct stages:

1) First, embeddings are generated within the same secure environment as the source database. This is a critical design choice, as it ensures that sensitive data is processed without requiring new access permissions.
2) Next, only the abstract embeddings are streamed (e.g., using a message queue like Kafka [7]) to the separate machine learning environment. The raw data never leaves its secure enclave.
3) Finally, models are trained on these embeddings, and the resulting endpoints are published as a service, allowing them to be used by applications while preserving privacy.

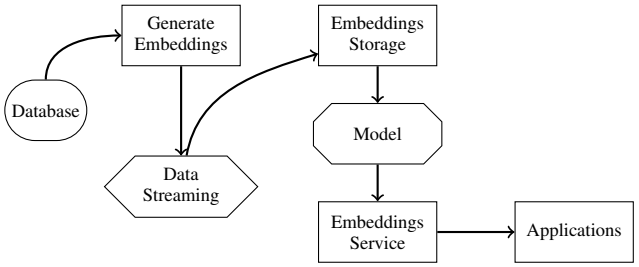

Fig. 1: Architecture diagram

The architecture diagram illustrates the high-level, complete end-to-end system for a production environment. This paper covers it only partially, focusing on validating the core components required to make such a system viable (i.e., generate embeddings, model training, embeddings service, and sample application). The implementation of the core Embedding Service is available on GitHub [8].

The following three sections (Data Pipeline, Embeddings Pipeline, and Model Framework and Architecture) describe each stage in details.

B. Data Pipeline

This study uses the publicly available Lending Club (LC) dataset [9], chosen specifically because it has undergone minor preprocessing. Other datasets, particularly in the credit fraud space where privacy concerns are of high importance, often obscure raw features through techniques like Principal Component Analysis (PCA) [10], the LC dataset provides the rich, mixed-type features (e.g., raw text, numerical values) that are essential for developing and testing a privacy-preserving technique that operates on original data. The only significant anonymization is the usage of k-anonymity [11] to mask the exact zip code, for example: 809xx.

The dataset contains records for approximately 2 million accepted loans. For this classification task, all loans with a "Current" status were filtered out, resulting in a final working dataset of approximately 1.4 million completed loans. Exploratory analysis revealed a significant class imbalance, with defaulted loans representing 21.2% of outcomes. This imbalance is addressed in the Model Training section.

A multistage feature selection process was employed to ensure methodological rigor. First, an initial pruning was performed, where features with over 60% missing values and those with high inter-feature correlation were removed, exceptions were made for pairs where both features demonstrated high predictive importance with the target variable, such as loan sub-grade and interest rate. Second, to ensure a realistic simulation of a real-world loan application scenario, any features containing "leaky" information (i.e., data that would not be available at the time of origination) were discarded.

TABLE I: SELECTED FEATURES AND DESCRIPTIONS

Feature Name	Description
`loan_amnt`	The total amount of the loan applied for by the borrower.
`int_rate`	The interest rate assigned to the loan.
`sub_grade`	The loan sub-grade assigned by Lending Club.
`title`	The loan title provided by the borrower (e.g., "Business").
`emp_title`	The job title provided by the borrower.
`emp_length`	The borrower's employment length in years.
`addr_state`	The state where the borrower resides.
`zip_code`	The first three digits of the borrower's zip code.
`dti`	Borrower's ratio of total monthly debt to monthly income.
`fico_range_high`	The upper bound of the borrower's FICO credit score.
`earliest_cr_line`	The month and year of the borrower's first credit line.
`revol_util`	Percent of revolving credit the borrower is using.
`mort_acc`	The number of mortgage accounts on record.
`avg_cur_bal`	The average current balance on all of the borrower's accounts.
`num_op_rev_tl`	The number of open revolving credit lines.

From the remaining 68 candidates, a final set of 15 features (shown on Table I) was selected based on a feature importance analysis using a preliminary XGBoost model [12] and a review of prior literature [13]–[15].

C. Embeddings Pipeline

This paper introduces a novel embeddings pipeline designed to unlock inaccessible private data for machine learning. The pipeline leverages the privacy-preserving nature of embeddings by transforming raw data into abstract, high-dimensional vectors. This transformation yields representations that are not directly human-interpretable, yet retain the essential relational patterns required for building powerful predictive models. Crucially, the proposed pipeline is specifically designed to handle the mixed-type datasets common in real-world records

by employing distinct components to process textual and numerical features.

The specific implementation of this pipeline for the experiments is detailed below.

For the experiments, a pre-trained sentence-transformer model was used to generate the embeddings. Given the large scale of the dataset, computational efficiency and cost were primary design constraints. Therefore, the all-MiniLM-L6-v2 (MiniLM) [16], [17] model was selected, which offers a strong balance between high embedding quality and efficient processing speed. (While more powerful models exist on the MTEB [18] and FinMTEB leaderboards [19], MiniLM represents a practical and robust choice.)

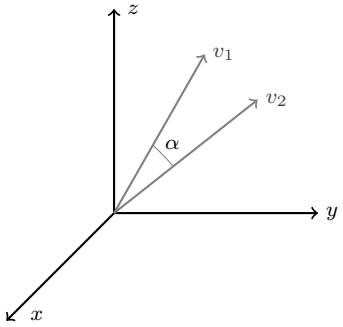

Fig. 2: Vector space.

While transformer-based embeddings models excel at capturing semantic meaning, they are not designed to preserve the mathematical properties of numerical data. To address this limitation, the pipeline explores incorporating a specialized technique for numeracy-preserving embeddings, Deterministic Independent of Corpus Embeddings (DICE) [20]. DICE transforms numbers such that the cosine similarity (1) between their vectors reflects their distance; the closer they are, the more likely for them to be similar (Figure 2).

$$\cos\alpha = \frac{v_1 \cdot v_2}{|v_1|\,|v_2|} \tag{1}$$

The length of the vectors v_1 and v_2 must be greater than zero ($|v_n| > 0$). Otherwise, a division by zero will lead to an undetermined value.

D. Model Framework and Architecture

To assess the performance of the proposed pipeline, a number of predictive models were considered based on their demonstrated success in prior analysis of the LC dataset. Consistent with the literature, XGBoost [21] and Neural Network [22] models were selected as subjects for this classification task. Given the sensitivity of these models to their configuration, an important goal was to ensure fairness across experiments. Thus, a consistent training and evaluation protocol was established. It was developed and validated on a subset of the data before being applied to full experiments.

To model the complex relationships within the data, the neural network used is a standard feed-forward network, a Multi-Layer Perceptron (MLP). The architecture consists of two hidden layers and a single output neuron suitable for a binary classification task. The design and parameter decisions are based upon that of the previous literature [22].

Consistent with common deep learning practices for high-dimensional data, the design employed a funnel structure [23]. The network's input layer is determined by the dimensionality of each data pipeline's feature vector (see Table II for a breakdown of feature vector sizes). The first hidden layer consists of 512 neurons, designed to learn a dense, compressed representation of the wide input. The second hidden layer further distills these patterns with 256 neurons. The final output layer consists of a single neuron to produce the classification logit. As an activation function, we selected the Rectified Linear Unit (ReLU) over older alternatives like tanh (see [22]), as it is a modern standard known for its computational efficiency and ability to mitigate the vanishing gradient problem [24].

1) Experimental Design: The experimental process was designed to isolate the impact of the embeddings pipeline by comparing three distinct data representation methods.

1) **Non-Private Baseline**: Models trained on the raw features. Textual features were processed using a TF-IDF vectorizer [25], and categorical features were one-hot encoded [26].
2) **Text-Only Embeddings**: Models trained on a dataset where all features, including numerical ones, were first cast to strings and then embedded using the all-MiniLM-L6-v2 transformer (384 dimensions).
3) **Hybrid Embeddings**: Models trained where text features were processed by the transformer and numerical features were processed using DICE embeddings (32 dimensions).

Each representation method results in a distinct input feature dimensionality, summarized in Table II.

TABLE II: INPUT VECTOR DIMENSIONS BY REPRESENTATION METHOD

Representation	Feature Composition	Input Size
Text-Only	15 features × 384-dim transformer embeddings	5760
Hybrid	(8 numerical × 32-dim DICE) + (7 text × 384-dim)	2944
Non-Private Baseline	Raw features (TF-IDF + one-hot encoded)	2514

2) Model Training: For model training and optimization, the data was partitioned into 3 sets for training (64%), validating (16%), and testing (20%). The intent is to ensure fair tuning and final evaluation. The validation set was utilized to implement early stopping for both models to prevent overfitting [27] and to determine the optimal number of training iterations (epochs). In addition, class imbalance was addressed by using a class weighting strategy to prevent either model from developing a bias toward the majority (negative) class. Thus, a parameter pos_weight was incorporated for both models. It is calculated as the ratio of the number of negative samples to the number of positive samples in the training data (2).

$$pos\_weight = \frac{\text{Number of negative samples}}{\text{Number of positive samples}} \quad (2)$$

This method increases the penalty for miss-classifying the minority (positive) class, pushing the models to learn the minority's distinguishing features more effectively, resulting in a more robust model.

The MLP was trained using the Adam optimizer, selected for its adaptive learning rate and robust performance, which aligns with findings in prior literature [22]. To handle the class imbalance present in the dataset, we used a BCEWithLogitLoss function, which combines a sigmoid activation with binary cross-entropy loss for numerical stability and allows for the addition of the pos_weight parameter.

To prevent overfitting, a major issue, and improve generalization, the training procedure incorporated two regularization techniques. First, Batch Normalization was applied after each linear transformation to stabilize and accelerate the training process. Second, a Dropout rate of 0.3 was applied to the hidden layers, forcing the network to learn more robust feature representations.

3) Evaluation Protocol: Model evaluation utilized a comprehensive set of metrics including Precision, Recall, F1-score [28], and the Area Under the Receiver Operating Characteristic Curve (ROC AUC) [29]. The primary metric for comparing the overall discriminative power of the different experimental pipelines (e.g., Non-Private Baseline vs. Hybrid Embeddings) was ROC AUC. Given the significant class imbalance, this threshold-independent metric provides the most robust measure of a model's intrinsic ability to distinguish between positive and negative classes.

Additionally, to provide insight into each model's practical utility at an optimal decision point, threshold-dependent metrics (Precision, Recall, and F1-score) are also reported. To ensure a fair comparison, these metrics were calculated using a classification threshold that was individually optimized for each model. This optimal threshold was determined by finding the point on the Precision-Recall curve that maximized the F1-score, thereby representing the model's best achievable balance between precision and recall.

III. Experimental Results

The following are the performance results of the XGBoost and Neural Network models across the three data representation pipelines, evaluated on the held-out test set. Detailed classification reports are in Tables IV and VI, and summaries are provided in Tables III and V.

A. XGBoost Performance

For the XGBoost model, the embedding-based pipelines demonstrated a modest but consistent improvement in discriminative ability over the non-private baseline. As summarized in Table III, the hybrid embedding pipeline achieved the highest performance with a ROC AUC score of 0.722. This surpassed both the Text-Only pipeline (0.721) and the Non-Private Baseline (0.717).

TABLE III: XGBOOST MODELS PERFORMANCE

Data Pipeline	ROC AUC Score
Non-Private Baseline	0.717
Text-Only Embeddings	0.721
Hybrid Embeddings	**0.722**

Although the ROC AUC scores indicate a stronger underlying signal from the embedded representations, this did not translate to significant gains in the threshold-dependent metrics. As shown in the detailed report in Table I, the F1-score for the positive (minority) class remained stable at 0.45 across all three pipelines. This suggests that while the model's ability to rank candidates improved with embeddings, the task of cleanly separating the classes at a single decision point remains challenging.

TABLE IV: XGBOOST MODELS CLASSIFICATION REPORT ON THE TEST SET

Data Pipeline	Class	Precision	Recall	F1-score	Support
Non-Private Baseline	Negative (0)	0.87	0.69	0.77	215748
	Positive (1)	0.35	0.63	0.45	58165
Text-Only Embeddings	Negative (0)	0.87	0.68	0.77	215748
	Positive (1)	0.35	0.64	0.45	58165
Hybrid Embeddings	Negative (0)	0.88	0.66	0.75	215748
	Positive (1)	0.34	0.67	0.45	58165

B. Neural Network Performance

In contrast to the XGBoost model, the Neural Network achieved its peak performance when trained on the Text-Only embeddings, yielding the highest ROC AUC of 0.718, outperforming both the hybrid pipeline (0.717) and the Non-Private Baseline (0.715), as shown in Table V. This suggests that the neural network, which processes all input features simultaneously, struggled to effectively integrate the two heterogeneous embedding types. In terms of threshold-dependent metrics like F1-score the Neural Network had stable performance across pipelines, similar to XGBoost.

TABLE V: NEURAL NETWORK MODELS PERFORMANCE SUMMARY

Data Pipeline	ROC AUC Score
Non-Private Baseline	0.715
Text-Only Embeddings	**0.718**
Hybrid Embeddings	0.717

The dense, 384-dimensional text embeddings contain a stronger and more complex signal than the 32-dimensional numerical DICE embeddings. For the neural network, concatenating these may have caused the less informative numerical features to act as noise, degrading the learning process and making it difficult for the network to capitalize on the patterns in the dominant text features.

Conversely, XGBoost's tree-based architecture is more robust to this feature heterogeneity. Its inherent feature selection

TABLE VI: NEURAL NETWORK MODELS CLASSIFICATION REPORT ON THE TEST SET

Data Pipeline	Class	Precision	Recall	F1-score	Support
Non-Private Baseline	Negative (0)	0.88	0.66	0.75	215748
	Positive (1)	0.34	0.66	0.45	58165
Text-Only Embeddings	Negative (0)	0.88	0.64	0.74	215748
	Positive (1)	0.34	0.67	0.45	58165
Hybrid Embeddings	Negative (0)	0.87	0.70	0.78	215748
	Positive (1)	0.36	0.61	0.45	58165

mechanism at each split allowed it to selectively leverage the most predictive features from either the text or numerical embeddings at any given decision point, effectively ignoring the noisier inputs when necessary.

IV. Conclusion

This research addresses the critical challenge of privacy in machine learning by introducing a pipeline that utilizes embeddings to enable model development on sensitive data. While the field of PPML has established powerful tools like Differential Privacy (DP) [30] and Homomorphic Encryption [31], these methods often involve well-documented trade-offs, such as significant utility loss or high computational overhead. This study explores a complementary approach, focusing on embeddings as a privacy-preserving foundation.

In conclusion, this paper demonstrates that embedding-based pipelines are a viable and effective strategy for building predictive models on sensitive data, enabling the development of privacy-preserving applications. This approach was validated by the highly competitive performance between the embedding-based models and the non-private baseline. The top-performing model, an XGBoost classifier using hybrid embeddings, achieved a ROC AUC score of 0.72, which is comparable to both our baseline and results from previous literature [21], [22].

This key finding suggests that abstracting raw data into high-dimensional embeddings successfully preserves the underlying predictive patterns necessary for effective modeling. A notable secondary result was the varied response of different model architectures to embedding strategies, highlighting the importance of matching the data representation to the chosen model.

The successful validation of this pipeline opens two key avenues for future work: enhancing model performance and formalizing privacy guarantees.

First, to improve predictive utility, more advanced embedding models need to be explored, such as state-of-the-art transformers from the MTEB leaderboard or domain-specific models like those on the FinMTEB leaderboard.

Second, a critical direction is to provide formal mathematical privacy guarantees. While it is known that applying Differential Privacy (DP) directly to raw, high-dimensional data can degrade model utility; the embedding pipeline presents a new opportunity. By analyzing the practical reversibility of embeddings to better understand the inherent privacy they provide [32]. A reduced amount of calibrated noise can be applied to the abstract embeddings rather than the raw features, making it possible to achieve a strong privacy guarantee with a much lower impact on model performance. Future work will investigate this trade-off, aiming to combine the privacy-preserving embedding approach with DP to create a framework that is both highly accurate and formally private.

References

[1] J. Joshi, R. Xu, and N. Baracaldo, "Privacy-Preserving Machine Learning: Methods, Challenges and Directions," arXiv preprint arXiv:2108.04417, 2021, [Online]. Available: https://doi.org/10.48550/arXiv.2108.04417

[2] G. Biau and B. Cadre, "Optimization by gradient boosting," arXiv eprint arXiv:1707.05023, 2017, [Online]. Available: https://doi.org/10.48550/arXiv.1707.05023

[3] M. Gabrié, S. Ganguli, C. Lucibello, and R. Zecchina, "Neural networks: from the perceptron to deep nets," arXiv eprint arXiv:2304.06636, 2023, [Online]. Available: https://doi.org/10.48550/arXiv.2304.06636

[4] P. Cunningham and S. Delany, "K-Nearest Neighbour Classifiers - A Tutorial," ACM Comput. Surv., vol. 54, New York, NY, USA, 2021, [Online]. Available: https://doi.org/10.1145/3459665

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela "Retrieval-augmented generation for knowledge-intensive NLP tasks," NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems, pp. 9459–9474, Vancouver, BC, Canada, 2020, [Online]. Available: https://dl.acm.org/doi/10.5555/3495724.3496517

[6] M. Yang, J. Chen, Y. Zhang, J. Liu, J. Zhang, Q. Ma, H. Verma, Q. Zhang, M. Zhou, I. King, and R. Ying "Low-Rank Adaptation for Foundation Models: A Comprehensive Review," arXiv, 2024, [Online]. Available: https://doi.org/10.48550/arXiv.2501.00365

[7] Apache Kafka, [Online]. Available: https://kafka.apache.org

[8] P. Buckner-Wolfson, "Embeddings-Service," Version 1.0.0, GitHub, 2025, [Online]. Available: https://github.com/PierceBW/Embeddings-Service

[9] Lending Club Dataset, [Online]. Available: https://www.kaggle.com/datasets/wordsforthewise/lending-club

[10] Credit Card Fraud Detection, [Online]. Available: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

[11] L. Sweeney, "k-anonymity: A model for protecting privacy," In: International Journal of Uncertainty, Fuzziness and Knowledge-based Systems, vol. 10, pp. 557–570. World Scientific Publishing Co., Inc. USA, 2002, [Online]. Available: https://doi.org/10.1142/S0218488502001648

[12] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," In: 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM, San Francisco, California, USA, 2016, [Online]. Available: https://doi.org/10.1145/2939672.2939785

[13] R. Emekter, Y. Tu, B. Jirasakuldech, and M. Lu, "Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending," Applied Economics **47**(1), 54–70, 2015, [Online]. Available: https://doi.org/10.1080/00036846.2014.962222

[14] N. Möllenkamp, "Determinants of loan performance in P2P lending," BS thesis, University of Twente, Enschede, The Netherlands (2017)

[15] M. Sharar, "Online Peer-to-Peer Lending: Determinants of Loan Performance," Available at SSRN: 3785323, 2021, [Online]. Available: https://doi.org/10.2139/ssrn.3785323

[16] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers," arXiv preprint arXiv:2002.10957, 2020, [Online]. Available: https://doi.org/10.48550/arXiv.2002.10957

[17] Sentence-Transformers, "all-MiniLM-L6-v2 model," [Online]. Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[18] K. Enevoldsen, I. Chung, I. Kerboua, M. Kardos, A. Mathur, D. Stap, et al., "MMTEB: Massive Multilingual Text Embedding Benchmark," arXiv preprint arXiv:2502.13595, 2025, [Online]. Available: https://doi.org/10.48550/arXiv.2502.13595

[19] Y. Tang and Y. Yang, "FinMTEB: Finance Massive Text Embedding Benchmark," arXiv preprint arXiv:2502.10990, 2025, [Online]. Available: https://doi.org/10.48550/arXiv.2502.10990

[20] L. Carin, D. Sundararaman, S. Si, V. Subramanian, G. Wang, and D. Hazarika, "Methods for Numeracy-Preserving Word Embeddings," In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 4742–4753. Association for Computational Linguistics, 2020, [Online]. Available: https://doi.org/10.18653/v1/2020.emnlp-main.384

[21] L. Souadda, A. Halitim, B. Benilles, J. Oliveira, and P. Ramos, "Optimizing Credit Risk Prediction for Peer-to-Peer Lending Using Machine Learning," Forecasting **7**(3), 35, 2025, [Online]. Available: https://doi.org/10.3390/forecast7030035

[22] J. Turiel and T. Aste, "Peer-to-Peer Loan Acceptance and Default Prediction with Artificial Intelligence," R. Soc. Open Sci. **7**(6), 191649, 2020, [Online]. Available: https://doi.org/10.1098/rsos.191649

[23] G. Hinton and R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science **313**(5786), 504–507, 2006, [Online]. Available: https://doi.org/10.1126/science.1127647

[24] A. Agarap "Deep Learning using Rectified Linear Units (ReLU)," arXiv preprint arXiv:1803.08375, 2019, [Online]. Available: https://doi.org/10.48550/arXiv.1803.08375

[25] "TF-IDF Demystified," https://towardsdatascience.com/tf-idf-demystified-7dd3ef071b24/, last accessed 2025/09/02

[26] "Robust One-Hot Encoding," [Online]. Available: https://towardsdatascience.com/robust-one-hot-encoding-930b5f8943af/

[27] "Overfitting," [Online]. Available: https://developers.google.com/machine-learning/crash-course/overfitting/overfitting

[28] "Understanding Precision, Recall, F1-score, and Support in Machine Learning Evaluation," [Online]. Available: https://medium.com/@nirajan.acharya777/understanding-precision-recall-f1-score-and-support-in-machine-learning-evaluation-7ec935e8512e

[29] "Area Under the Curve (AUC): A Robust Performance Measure of Classification Models," [Online]. Available: https://medium.com/@bayramorkunor/area-under-the-curve-auc-a-robust-performance-measure-of-classification-models-cbfc3549d8c6

[30] R. Danger, "Differential Privacy: What is all the noise about?" arXiv preprint arXiv:2205.09453, 2022, [Online]. Available: https://doi.org/10.48550/arXiv.2205.09453

[31] J. Agrawal and H. Nguyen, "Fraud Detection Using Machine Learning Models and Encryption Techniques," In: H. Arabnia, L. Deligiannidis, S. Amirian, et al. (eds.) "Artificial Intelligence and Applications," pp. 115–124. Springer Nature Switzerland, Cham, 2025, [Online]. Available: https://doi.org/10.1007/978-3-031-86623-4_9

[32] J. Morris, V. Kuleshov, V. Shmatikov, and A. Rush, "Text Embeddings Reveal (Almost) As Much As Text," arXiv preprint arXiv:2310.06816, 2023, [Online]. Available: https://doi.org/10.48550/arXiv.2310.06816