Integrated Deep Learning Strategy for Hybrid Credit Risk Prediction and Text Classification Tasks

K Bhaskar[1], S Mohan Krishna[2], K Keerthana[3]
[1]Assistant Professor, Department of CSE(AI & ML), Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: bhaskark.mca@gmail.com , ORC-ID: https://orcid.org/0009-0000-3309-4240
[2]P.G Scholar, Department of MCA, Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: prabhaprabhas2727@gmail.com , ORC-ID: https://orcid.org/0009-0008-4341-3055
[3]Assistant Professor, Department of CSE(AI & ML), Sri Venkatesa Perumal College of Engineering & Technology, Puttur, E-mail: keerthanakalasamudram17@gmail.com , ORC-ID: https://orcid.org/0009-0009-1485-1731

To Cite this Article
K Bhaskar, S Mohan Krishna, K Keerthana, "Integrated Deep Learning Strategy for Hybrid Credit Risk Prediction and Text Classification Tasks", Journal of Science Engineering Technology and Management Science, Vol. 03, Issue 04, April 2026,pp: 319-331, DOI: http://doi.org/10.64771/jsetms.2026.v03.i04.pp319-331
Submitted: 28-02-2026 Accepted: 01-04-2026 Published: 08-04-2026

Abstract: For modern financial analytics to work, credit scoring needs to be strong across both organized and unstructured data. This is because traditional statistical and shallow learning methods don't work well with complex non-linear, temporal, and semantic dependencies. A unified system based on deep learning is created for both categorizing text and predicting numerical credit risk. A public credit risk dataset and standard text datasets (News, SciHTC, and UTCD) are used in the experiments. A common pipeline for preprocessing, visualization, and encoding is used, with changes made based on the job. We use contextual transformer embeddings to model text data and compare basic neural classifiers with a hybrid loss-driven LSTM and an ELECTRA-based design to improve how the data represents meaning. For numerical prediction, recursive feature elimination is the best way to choose features, and SMOTE helps even out class mismatch. We test basic machine learning models against a mixed LSTM and a soft-voting ensemble that combines XGBoost and LightGBM. Accuracy, precision, recall, F1-score, and AUC are some performance measures. The ensemble is 95.7% accurate at predicting credit risk, and ELECTRA is 96.1%, 97.3%, and 99.3% accurate on News, SciHTC, and UTCD. LIME and SHAP make the predictions easy to understand, and a Flask-based interface lets people guess and interact with the models in real time.

"Index Terms: *Credit scoring, deep learning, LSTM, transformer embeddings, explainable AI, financial risk prediction".*

1. INTRODUCTION

Credit scoring is an important part of today's financial systems; it affects how loans are made, how risks are managed, and how regulations are followed. Adding AI to credit evaluation has increased the amount of data that can be analyzed, which lets institutions move away from basic rule-based evaluations and toward data-driven decision support systems [1]. More and more alternative and behavioral data, like digital and social information, is becoming available. This has changed credit analytics even more by making it easier for more people to get credit [2]. At the same time, data mining and other advanced analytical techniques are being used more and more to improve the accuracy of predictions and deal with the problems that come up with the distribution of data in financial records [3]. The way intelligent credit assessment systems work is changing because of these new advances.

Even with these improvements, there are still big problems with handling different kinds of financial data in a way that is fair and easy to understand. Domain adaptation and explainability have become important factors in internet finance, but they are still hard to incorporate into unified credit score systems [4]. Also, cost-sensitive and selective ensemble methods show better accuracy in predictions, but they make it harder to balance performance and transparency [5]. Along with numerical data, unstructured written and behavioral data also causes problems. Surveys on deep learning-based text classification stress the value of advanced representation learning while also pointing out problems with generalization and stability [6, 7]. Attacker flaws and changes in the text modeling domain make it even harder to apply securely in secure financial settings [8].

This study suggests a unified and easy-to-understand approach for credit scoring that combines structured numerical data with unstructured behavioral data in a scalable analytical architecture. This will help with these problems. The system is meant to improve the accuracy of predictions, the ability to generalize across different types of data, and the openness of automated decision-making. The method aims to close the gap between predicting performance and being accountable to regulators by including features that allow fairness-aware modeling and outputs that can be explained. In addition, the approach is in line with improvements in transfer learning and contextual representation methods that have been shown to work well in difficult classification situations [9, 10].

The important thing about this work is that it could improve automatic credit checks while also encouraging fair and open financial practices. The suggested framework helps make risk assessment more accurate and builds trust between institutions by addressing problems like imbalance, heterogeneity, and interpretability. In the end, the study supports the creation of responsible, data-driven financial decision systems that can work well in real-life situations that are always changing.

2. LITERATURE REVIEW

Recent progress in classifying texts puts more and more emphasis on modeling papers' structures and relationships. Zhang et al. [11] suggested an inductive approach based on graph neural networks and showed that structural information at the document level can greatly improve classification performance. Chen et al. [12] combined contextual embeddings with convolutional architectures to improve the semantic representation for long texts when they used them for long-text classification. In the same way, Chen et al. [13] created a semi-supervised interpolation approach in hidden space to improve generalization when there isn't a lot of labeled data. These methods make representation learning and adaptability better, but they only work on textual topics and don't address the need to integrate with different types of data sources or make sure that the representation can be understood.

More attention has also been paid to hierarchical modeling as a way to better show label relationships and document structures. Wang et al. [14] used contrastive learning to add hierarchical information to encoder representations to make them stronger. Ding et al. [15] suggested hypergraph attention networks to model the more complex links between documents and labels. To use both semantic and relational cues, Lin et al. [16] mixed contextual embeddings with graph convolutional networks. To improve performance in organized label spaces, Zhou et al. [17] created a global framework that is aware of hierarchies. Even though these methods are very good at making predictions in difficult classification tasks, they often make architectures more complicated and increase the cost of computing, which makes them hard to scale and use in real-world financial systems.

Newer studies look into large-scale pre-training and prompt-based methods. Hu et al. [18] improved classification robustness by using outside information in prompt tuning strategies. They saw improvements in domain-specific situations. Sun et al. [19] looked at how large language models can be used for text classification. They talked about how good their zero-shot and few-shot skills are, but they also raised concerns about how well they work and how well they adapt to new domains. Wen and Fang [20] solved the problem of not having enough resources by combining graph-grounded pre-training with prompting mechanisms. This led to better performance even when there wasn't a lot of labeled data. Even though these methods work, they often need a lot of computing power and are mostly focused on textual analysis, not multimodal financial information or problems with how they can be interpreted.

Overall, the study that has been done so far shows that structural modeling, hierarchical learning, and prompt-based adaptation for text classification have come a long way. Still, there are some problems that need to be fixed. For example, it's not easy to combine different types of structured and unstructured data, there are worries about how to make the system work on a larger scale, and there isn't enough focus on making decisions that are clear in high-stakes areas like credit scores. Also, a lot of methods focus on how well they classify things without really thinking about how fair, strong, or generalizable they are across domains in financial settings. These gaps led to this study's goal of creating a unified and easy-to-understand system that can combine different types of data while still being able to make accurate predictions and be used in real life.

3. MATERIALS AND METHODS

The suggested system creates a unified deep learning framework for textual classification and numerical credit risk prediction. It does this by using a real-world credit risk dataset along with standard datasets from News, SciHTC, and UTCD. Textual inputs are encoded using contextual transformer embeddings, such as BERT, RoBERTa, and ELECTRA, to capture semantic dependencies. This is done by following standard processes for preparing and partitioning data. Several neural designs are tested along with a hybrid loss–driven LSTM model to make the classification more reliable. For numerical prediction, recursive feature elimination finds characteristics that make a difference, and SMOTE reduces class imbalance to make generalization better. A soft-voting ensemble that includes XGBoost and LightGBM also uses their complementary learning skills to lower variance and improve the stability of predictions. Confusion matrices and metrics like accuracy, precision, recall, F1-score, and AUC are used to measure how well a model works. Explainable AI methods, like LIME and SHAP, offer clear feature-level interpretation. A Flask-based deployment framework allows for real-time inference and interactive access, which makes the system scalable, reliable, and useful in financial decision-support settings.

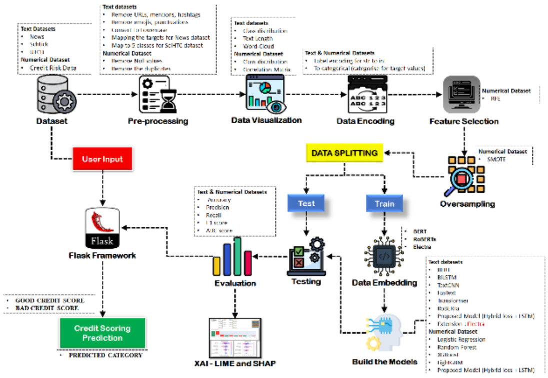


Fig.1 Proposed Architecture

There are layers for preprocessing data, extracting features, training models, evaluating them, and deploying them all built into a single structure. Before being processed by machine learning and deep learning models, textual and numerical data are cleaned, normalized, and encoded. To make sure that the system is reliable and strong, performance evaluation modules calculate classification measures. The improved models are put into use through a Flask-based web interface, which lets users make predictions and interact with the models in real time. An integrated explainability component gives outputs that can be understood, which improves system trustworthiness, openness, and decision support.

a) Dataset Collection:

The Credit Risk dataset comes from Kaggle and has 12 attributes that describe 32,581 loan application records. These attributes include demographic, financial, job, and credit history information. It has both numeric and categorical factors, and the binary target label loan_status is used to identify them. The dataset has missing values and different types of features that represent how finances work in the real world. This makes it a good source for accurate modeling and evaluating credit risk.

	person_age	person_income	person_home_ownership	person_emp_length	loan_intent	loan_grade	loan_amnt	loan_int_rate	loan_status	loan_percent_income	cb_person_default_on_file	cb_person_cred_hist_length
0	22	59000	RENT	123.0	PERSONAL	D	35000	16.02	1	0.59	Y	3
1	21	9600	OWN	5.0	EDUCATION	B	1000	11.14	0	0.10	N	2
2	25	9600	MORTGAGE	1.0	MEDICAL	C	5500	12.87	1	0.57	N	3
3	23	65500	RENT	4.0	MEDICAL	C	35000	15.23	1	0.53	N	2
4	24	54400	RENT	8.0	MEDICAL	C	35000	14.27	1	0.55	Y	4

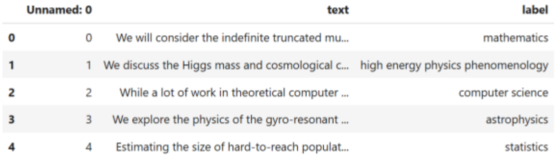
Fig.2 Credit Risk

The AG News dataset comes from Kaggle and is made up of 127,600 news stories that were put together from training and testing sets that were already set up. Each record has a news story description in text form and an integer label that stands for one of four groups. The dataset is made up of structured pairs of text and labels that don't have any missing values. This gives it a balanced class distribution and broad field coverage. It can be used to test text classification models and see how well they learn semantic representations because it is big, has clean annotations, and is structured in multiple classes.

	text	label
0	Wall St. Bears Claw Back Into the Black (Reute...	2
1	Carlyle Looks Toward Commercial Aerospace (Reu...	2
2	Oil and Economy Cloud Stocks' Outlook (Reuters...	2
3	Iraq Halts Oil Exports from Main Southern Pipe...	2
4	Oil prices soar to all-time record, posing new...	2

Fig.3 AG News

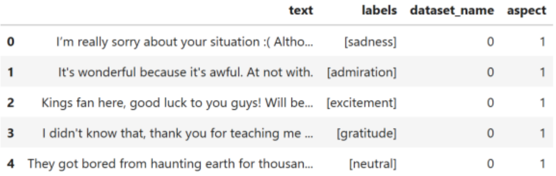
The Scientific Text Classification dataset comes from Kaggle and is made up of 78,631 study abstracts that have been labeled in a number of different scientific areas, such as computer science, astrophysics, statistics, and high-energy physics. Each record has writing and a label that describes what it is. The large-scale, domain-diverse collection helps to test and improve multi-class text classification models and semantic representation learning in specific academic settings.



	Unnamed: 0	text	label
0	0	We will consider the indefinite truncated mu...	mathematics
1	1	We discuss the Higgs mass and cosmological c...	high energy physics phenomenology
2	2	While a lot of work in theoretical computer ...	computer science
3	3	We explore the physics of the gyro-resonant ...	astrophysics
4	4	Estimating the size of hard-to-reach populat...	statistics

Fig.4 SciHTC

The Hugging Face repository has the Unified Text Classification Dataset (UTCD), which has 168,365 text examples that have been labeled. There is textual material, emotion-based labels, and extra metadata fields in each instance. The large-scale, multi-label structure with a wide range of expressive categories makes it easier to test contextual embedding models and improves generalization across a wide range of semantic and emotional text classification tasks.



	text	labels	dataset_name	aspect
0	I'm really sorry about your situation :(Altho...	[sadness]	0	1
1	It's wonderful because it's awful. At not with.	[admiration]	0	1
2	Kings fan here, good luck to you guys! Will be...	[excitement]	0	1
3	I didn't know that, thank you for teaching me ...	[gratitude]	0	1
4	They got bored from haunting earth for thousan...	[neutral]	0	1

Fig.5 UTCD

b) Pre-Processing:

The data preparation step creates a structured base for modeling by making sure that the data is correct, consistent, and useful for representation. This improves the stability of learning, the accuracy of predictions, and the overall robustness of the system across a wide range of datasets.

i. Data Processing: Both textual and numerical entries are standardized during data processing to make sure that results are consistent and reliable. Some of the tasks that are done during cleaning are getting rid of noise, normalizing the content, mapping target labels to records, getting rid of lost records, and filtering out duplicates. These steps improve the accuracy of the data and lower the variation that comes from mistakes or missing records. Pre-processing makes the data better overall, which strengthens model convergence, reduces training instability, and sets a solid basis for later stages of analysis within the unified predictive framework.

ii. Data Visualization: Before modeling, data visualization is used to look at the characteristics of a dataset and find structural trends. Analytical plots like class distribution graphs, text-length analysis, word clouds, and correlation matrices show how features are related to each other and how they are imbalanced. These visual evaluations help people make smart choices about sampling methods and feature engineering. Visualization makes data easier to understand by showing secret patterns and possible biases. It also makes sure that modeling decisions are based on facts, not assumptions.

iii. Encoding Strategy: To make them work with machine learning and deep learning frameworks, categorical variables and goal labels are turned into structured numerical representations. This encoding method keeps the semantic meaning while turning discrete characteristics into formats that can be analyzed. The right way to show category data makes sure that parameters are learned quickly and clearly, and it stops confusion during model training. The encoding approach makes computations more stable and improves overall predictive consistency by standardizing input formats across different datasets.

iv. Feature Selection: The numerical dataset is put through Recursive Feature Elimination to find the most useful financial characteristics that help predict credit risk. This method cuts down on the number of dimensions and gets rid of weak or unnecessary predictors by checking the usefulness of features over and over again. Focusing on distinguishing traits speeds up computations and makes it easier for models to be used in other situations. Choosing the right features also lowers the risk of overfitting, making sure that the success of the prediction is based on useful financial indicators instead of noise or false correlations.

v. Oversampling Technique: The Synthetic Minority Oversampling Technique is used to fix the class mismatch in the credit risk dataset. This method creates synthetic samples that are representative of underrepresented groups.

This makes the training distribution more even. Balanced data make classifiers more sensitive to cases that aren't in the majority and less biased toward forecasts that are in the majority. This makes recall and general robustness better, which helps make credit risk assessment more fair and reliable in real-world financial environments that aren't always balanced.

vi. Data Splitting: To split datasets into training and testing groups while keeping the original class proportions, stratified data partitioning is used. Keeping the distributional stability across splits makes sure that performance reviews are fair and stops sampling bias. This approach makes it easier to repeat results and lets models be compared objectively. The framework makes sure that reported performance metrics accurately show generalization ability instead of memorization by separating the training and evaluation stages in the right way.

vii. Textual Embedding Generation: Transformer-based models like BERT, RoBERTa, and ELECTRA are used for textual embedding creation, which turns raw text into dense contextual representations. Beyond what you can see, these embeddings record semantic links, contextual dependencies, and linguistic subtleties. This method lets neural architectures quickly learn important patterns by storing text in a high-dimensional semantic space. Contextual embeddings make classification much more accurate and reliable across a wide range of textual areas.

c) Algorithms:

For Textual Dataset Models:

BERT: BERT creates deep contextual embeddings by modeling how text sequences rely on each other in both directions. Its attention system picks up on semantic relationships and long-range interactions, which makes it better at classifying things [21] and knowing what's going on around it. Through rich pre-trained representations, the model improves its ability to generalize and be robust across a wide range of linguistic patterns.

BiLSTM: BiLSTM works with sequences both forwards and backwards to find relationships in time and space. This two-way modeling makes it easier to see sequential trends and keeps information from getting lost. Its structure makes learning steady over long texts, which improves the consistency and generalization of classification [22].

TextCNN: Convolutional filters are used by TextCNN to pull out important local features and n-gram patterns from text embeddings. Pooling processes improves consistency and speed up computation. The design provides strong performance with less complexity, which allows for stable and accurate text classification [23].

FastText: FastText uses character-level n-grams to improve how words are represented, which makes it better at handling uncommon words and morphological changes. It does good semantic modeling [24] by combining knowledge about words and their parts. Its light design lets you train quickly while keeping up with the best predicted performance.

Transformer: Self-attention mechanisms are used by transformer designs to model global contextual dependencies that don't happen again and again. This parallel structure makes it easier to scale up and effectively records interactions over long distances [25]. The method improves the quality of semantic representation and helps with strong generalization in difficult text classification tasks.

RoBERTa: RoBERTa improves language models based on transformers by using better pre-training methods and dynamic masking. It makes strong contextual embeddings that can capture complex semantic links. The better quality portrayal makes classification more accurate and less affected by differences in language [26].

Hybrid Loss–Driven LSTM: To find the right mix between discrimination and generalization, Hybrid Loss–Driven LSTM combines sequential modeling with optimized loss functions. The design takes into account temporal dependencies and improves the stability of convergence. This design makes it more resistant to class imbalance and overfitting, which means that forecasts can be trusted.

ELECTRA: ELECTRA uses a pre-training method based on discriminators to learn how to describe context through token-level replacement detection. This effective way of teaching improves semantic differentiation with less work for the computer. The embeddings that are made improve the accuracy of segmentation and the ability to scale [27].

For Models of Numeric Datasets:

For Numeric Dataset Models:

Logistic Regression: LR is a probabilistic baseline classifier that models how features and result probabilities are related in a straight line [28]. Its structure can be understood, and its regularization features make it steady, giving us reliable standards for comparing things in risky situations.

$$P(y = 1\,|X) = \frac{1}{1 + e^{-(W^T x + b)}} \qquad (1)$$

Random Forest: RF combines several decision trees that were trained on randomized groups to lower variance and improve robustness. It does a good job of capturing nonlinear feature relationships [29] while preventing overfitting. The ensemble structure makes predictions more stable and helps with figuring out which features are most important.

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \qquad (2)$$

Extreme Gradient Boosting: XGBoost builds sequential decision trees that are better thanks to gradient-based learning. It gets high predicted accuracy and strong generalization by fixing mistakes over and over again. For structured data classification jobs [30], regularization and efficient computation make them more robust.

$$\hat{y}_i = \sigma\left(\sum_{k=1}^{K} f_k(x_i)\right), f_k \in F \qquad (3)$$

Light Gradient Boosting Machine: LightGBM makes gradient boosting work better by using histogram-based learning and tree growth step by step. This system speeds up training without affecting the quality. It can handle large amounts of data and can be scaled up or down, which helps it make accurate number predictions.

Soft Voting Classifier: The Ensemble Model takes the probabilistic results from several models and puts them all together to make estimates that everyone agrees on. It lowers bias and variance by combining skills that work well together. Probability-based fusion improves the reliability, stability, and general confidence in classification in situations where decisions are hard to make.

$$\hat{y} = argmax_c \left(\sum_{i=1}^{n} II\ (\hat{y}_i = c)\right) \qquad (4)$$

d) Integration of XAI with Flask framework:

The combination of Explainable Artificial Intelligence (XAI) with the Flask framework creates a clear and understandable setting for deploying predictive models. Flask is a simple framework for building web apps. It handles routing, user input, model invocation, and answer rendering. The backend holds the learned machine learning and deep learning models, which lets structured HTTP requests do inference in real time. This integration makes sure that the user interface and the predictive engine can talk to each other without any problems. It also keeps the modular design and the ability to deploy in different sizes.

The XAI part makes the model more clear by giving us results that we can understand, like confidence levels, feature importance scores, keyword highlights, and decision summaries. Along with the prediction results, these descriptions are processed and dynamically shown using Flask templates. The system makes decision-support applications more trustworthy by combining inference and interpretability into a single structure. This increases user trust, analytical clarity, and practical dependability.

4. EXPERIMENTAL RESULTS

Accuracy: How well a test can tell the difference between sick and healthy people is called its accuracy. To get an idea of how accurate a test is, we should figure out what percentage of cases are true positives and true negatives. In terms of math, this can be written as

$$Accuracy = \frac{\mathrm{TP + TN}}{\mathrm{TP + FP + TN + FN}} (5)$$

Precision: Precision is the percentage of correctly classified cases or samples compared to those that were correctly classified as positives. So, here is the method to figure out the precision:

$$Precision = \frac{\text{True Positive}}{\text{True Positive } + \text{ False Positive}} (6)$$

Recall: In machine learning, recall is a metric that shows how well a model can find all the important instances of a certain class. It shows how well a model captures instances of a certain class. It is calculated by dividing the number of correctly predicted positive observations by the total number of real positives.

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP\ +\ FN}} (7)$$

F1-Score: The F1 score is a way to rate the correctness of a machine learning model. It takes a model's accuracy and recall scores and adds them together. The accuracy metric counts how many times, across the whole dataset, a model made a correct guess.

$$F1\,Score = 2 * \frac{Recall \text{ X } Precision}{Recall + Precision} * 100 \quad (8)$$

AUC-ROC Curve: The AUC-ROC Curve shows how well a classification problem is solved at different benchmark levels. The True Positive Rate is plotted against the False Positive Rate by ROC. AUC measures how well the model can tell the difference between classes; a higher AUC means the model works better.

$$AUC = \sum_{i=1}^{n-1} (FPR_{i+1} - FPR_i) \cdot \frac{TPR_{i+1} + TPR_i}{2} \quad (9)$$

Table.1 Performance Evaluation – Credit Risk

ML Model	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	0.736	0.739	0.737	0.736	0.816
Random Forest	0.926	0.927	0.926	0.926	0.979
XGBoost	0.946	0.949	0.946	0.946	0.984
LightGBM	0.947	0.950	0.947	0.947	0.983
Voting	**0.957**	**0.958**	**0.957**	**0.957**	**0.988**
Proposed LSTM + Hybrid Loss	0.840	0.854	0.840	0.839	0.924

Table.1 shows that the Voting classifier has the best predictive performance, with the highest overall accuracy, the ability to tell the difference between things, and the most balanced classification efficiency.

Table.2 Performance Evaluation – News

ML Model	Accuracy	Precision	Recall	F1 Score	AUC
BERT	0.935	0.936	0.935	0.935	0.990
BiLSTM	0.913	0.913	0.913	0.913	0.984
TextCNN	0.897	0.898	0.897	0.897	0.979
FastText	0.917	0.917	0.917	0.917	0.985
Transformer	0.896	0.898	0.896	0.896	0.978
RoBERTa	0.936	0.936	0.936	0.936	0.991
Proposed Model	0.902	0.902	0.901	0.901	0.983
Extension ELECTRA	**0.961**	**0.961**	**0.961**	**0.961**	**0.995**

Table.2 shows that Extension ELECTRA has the best total accuracy, precision, recall, F1-score, and AUC, showing that it is better at both generalization and classification.

Table.3 Performance Evaluation – SciHTC

ML Model	Accuracy	Precision	Recall	F1 Score	AUC
BERT	0.867	0.765	0.752	0.758	0.968
BiLSTM	0.850	0.735	0.680	0.700	0.953
TextCNN	0.841	0.729	0.665	0.687	0.940
FastText	0.864	0.770	0.707	0.731	0.965
Transformer	0.811	0.693	0.625	0.640	0.944
RoBERTa	0.874	0.777	0.763	0.769	0.972

Proposed Model	0.839	0.649	0.593	0.614	0.959
Extension ELECTRA	**0.973**	**0.973**	**0.972**	**0.972**	**0.997**

Table.3 shows that Extension ELECTRA has the best accuracy and discrimination for predictions, and it also has the best overall classification stability and reliability among the approaches that were tested.

Table.4 Performance Evaluation – UTCD

ML Model	Accuracy	Precision	Recall	F1 Score	AUC
BERT	0.970	0.955	0.919	0.936	0.993
BiLSTM	0.947	0.904	0.876	0.888	0.982
TextCNN	0.944	0.894	0.872	0.883	0.979
FastText	0.945	0.909	0.857	0.881	0.983
Transformer	0.919	0.829	0.829	0.828	0.971
RoBERTa	0.973	0.944	0.943	0.944	0.995
Proposed Model	0.954	0.920	0.876	0.897	0.989
Extension ELECTRA	**0.993**	**0.993**	**0.993**	**0.993**	**1.000**

The results shown in Table.4 show that Extension ELECTRA had the best overall predictive performance across all classification measures that were tested.

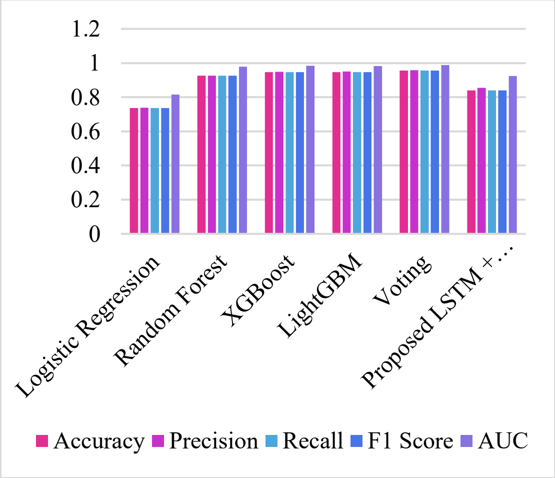


Fig.6 Comparison Graph – Credit Risk

Fig.6 shows a comparison of evaluation metrics across models, clearly showing that the Voting classifier is better at overall accuracy and balanced predictive effectiveness. This shows a consistent performance edge in the form of a graph.

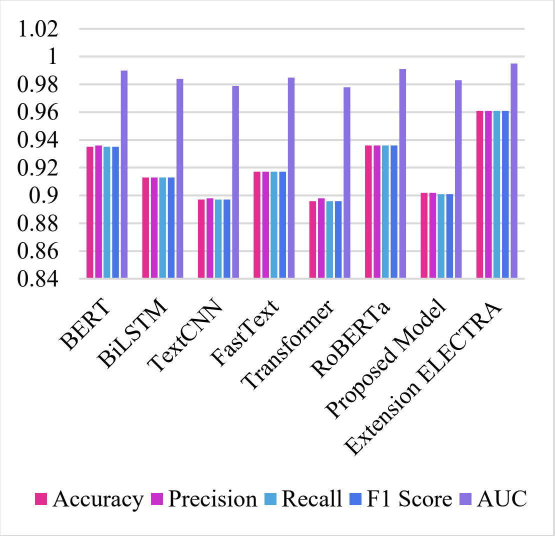


Fig.7 Comparison Graph – News

Fig.7 shows the comparison results, showing that Extension ELECTRA is clearly better than the other options, showing stronger generalization and better classification consistency across all criteria that were tested.

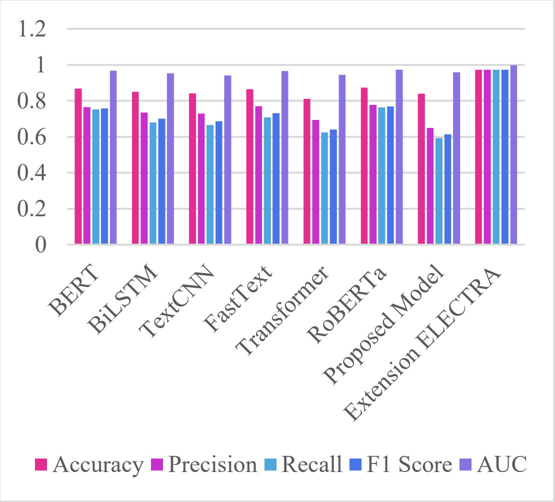


Fig.8 Comparison Graph – SciHTC

Figure 8 shows performance trends that show Extension ELECTRA's reliability while it is being evaluated. It has much higher stability and discrimination capabilities than competing approaches.

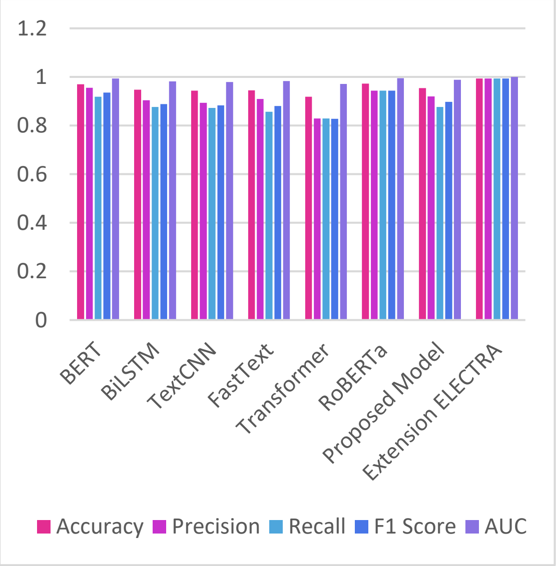
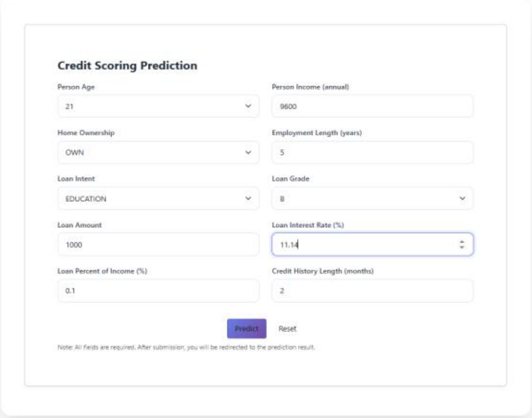


Fig.9 Comparison Graph – UTCD

In Fig.9, you can see a nearly perfect measure distribution for Extension ELECTRA. This shows how accurate the predictions are and how reliable the classifications are.



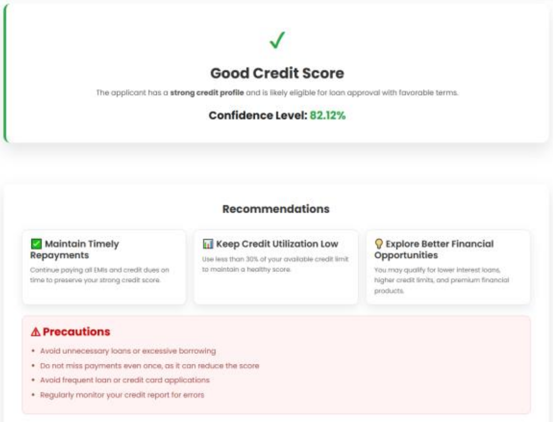
Credit Scoring Prediction

Person Age: 21
Person Income (annual): 9600
Home Ownership: OWN
Employment Length (years): 5
Loan Intent: EDUCATION
Loan Grade: B
Loan Amount: 1000
Loan Interest Rate (%): 11.14
Loan Percent of Income (%): 0.1
Credit History Length (months): 2

Note: All fields are required. After submission, you will be redirected to the prediction result.

Fig.10 Enter Input Data

Figure 10 shows an interface for predicting credit scores. It has structured input fields for personal, job, income, and loan information, as well as controls for sending the information.



✓

Good Credit Score

The applicant has a **strong credit profile** and is likely eligible for loan approval with favorable terms.

Confidence Level: 82.12%

Recommendations

✅ Maintain Timely Repayments
Continue paying all EMIs and credit dues on time to preserve your strong credit score.

📊 Keep Credit Utilization Low
Use less than 30% of your available credit limit to maintain a healthy score.

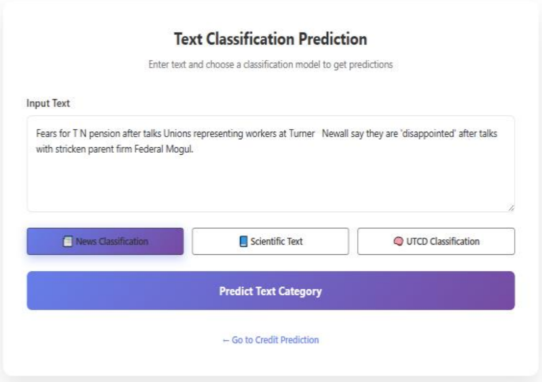
💡 Explore Better Financial Opportunities
You may qualify for lower interest loans, higher credit limits, and premium financial products.

⚠ Precautions

- Avoid unnecessary loans or excessive borrowing
- Do not miss payments even once, as it can reduce the score
- Avoid frequent loan or credit card applications
- Regularly monitor your credit report for errors

Fig.11 Predicted Results

Figure 11 shows the prediction results, which show a person's good credit score, trust level, personalized suggestions, and safety advice for keeping their finances stable.



Text Classification Prediction

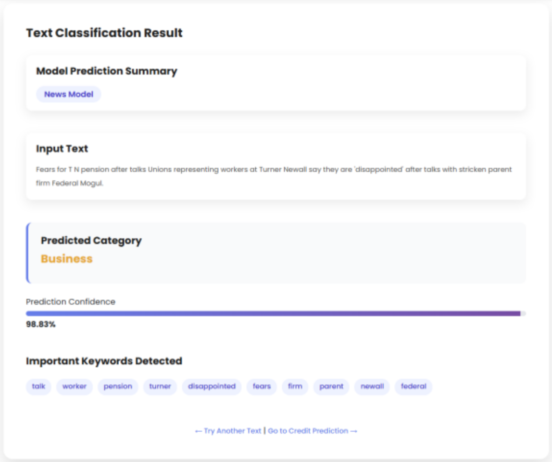
Enter text and choose a classification model to get predictions

Input Text

Fears for T N pension after talks Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.

Fig.12 Enter Input Text

Figure 12 shows a text classification interface with interactive control buttons that let you enter text, choose a model, and guess the group.



Text Classification Result

Model Prediction Summary

News Model

Input Text

Fears for T N pension after talks Unions representing workers at Turner Newall say they are 'disappointed' after talks with stricken parent firm Federal Mogul.

Predicted Category

Business

Prediction Confidence

98.83%

Important Keywords Detected

talk worker pension turner disappointed fears firm parent newall federal

Fig.13 Text Classification Result

Figure 13 shows the results of classifying text. It shows the predicted category, the confidence percentage, the input summary, and the keywords that were found to support the choice.

5. CONCLUSION

This system was created to meet the growing need for accurate credit score and text classification across a wide range of different financial data sources, including both structured numerical attributes and unstructured textual data. A combined deep learning framework was created using three benchmark text datasets: News, SciHTC, and UTCD. A freely available numerical credit risk dataset was also used. The method uses a standard pipeline for preparation, visualization, and encoding. It then uses contextual transformer embeddings to teach textual representation. A hybrid loss–driven LSTM design and an ELECTRA-based model were tested with a number of baseline neural text classifiers to improve semantic differentiation. To find discriminative features for numerical credit risk forecast, recursive feature elimination was used. To fix class imbalance, SMOTE was used. A hybrid LSTM model and a soft-voting ensemble made up of XGBoost and LightGBM were put up against classical machine learning algorithms. Accuracy, precision, recall, F1-score, and AUC were used to measure how well the model worked. The voting ensemble is able to accurately identify numerical credit risk 95.7% of the time. For ELECTRA-based text classification, it is able to achieve accuracy of 96.1% on News datasets, 97.3% on SciHTC datasets, and 99.3% on UTCD datasets. To make things clearer and more reliable, explainable AI methods based on LIME and SHAP were added, which allowed for feature-level interpretation. A Flask-based interface also makes it easier for users to connect and deploy in real time. Overall, the system that was made has good prediction performance, is easier to understand, and can be used in real life. It helps computers make smart decisions in areas like financial analytics and risk assessment.

Adding multimodal data sources like transaction sequences, customer demographics, and other behavioral cues can make the system even better and make predictions more accurate. To make generalization better across changing financial situations, researchers may look into more advanced self-supervised and domain-adaptive transformer models. To deal with idea drift in real-time credit risk assessment, online and incremental learning tools can be combined. Spread out training and deploying in the cloud for big financial environments can make scalability stronger. Adding more causal explainability methods and regulatory-compliant interpretability frameworks can also help make automated credit decision-making systems more open, fair, and trustworthy.

REFERENCES

[1] Mohanty, A. (2025). Artificial Intelligence in Credit Scoring: Enhancing Financial Inclusion & Opportunities.

[2] Alamsyah, A., Hafidh, A. A., & Mulya, A. D. (2025). Innovative Credit Risk Assessment: Leveraging Social Media Data for Inclusive Credit Scoring in Indonesia's Fintech Sector. Journal of Risk and Financial Management, 18(2), 74.

[3] Lakshmi, J. M., Prasad, K. K., & Viswanath, G. (2025). Proactive Security in Multi-Cloud Environments: A Blockchain Integrated Real-Time Anomaly Detection and Mitigation Framework. Cuestiones De Fisioterapia, 54(2), 392–417.

[4] Xu, F., & Zhang, R. (2025). Explainable domain adaptation learning framework for credit scoring in internet finance through adversarial transfer learning and ensemble fusion model. Mathematics, 13(7), 1045.

[5] Xiao, J., Li, S., Tian, Y., Huang, J., Jiang, X., & Wang, S. (2025). Example dependent cost sensitive learning based selective deep ensemble model for customer credit scoring. Scientific Reports, 15(1), 6000.

[6] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad-Khasmakhi, M. Chenaghlu, and J. Gao, "Deep learning-based text classification," ACM Comput. Surveys, vol. 54, no. 3, pp. 1–40, 2021.

[7] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "A survey on text classification algorithms: From text to predictions," Information, vol. 13, no. 2, p. 83, Feb. 2022.

[8] S. Garg and G. Ramakrishnan, "BAE: BERT-based adversarial examples for text classification," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020.

[9] R. Qasim, W. H. Bangyal, M. A. Alqarni, and A. Ali Almazroi, "A fine-tuned BERT-based transfer learning approach for text classification," J. Healthcare Eng., vol. 2022, pp. 1–17, Jan. 2022.

[10] V. Dogra, S. Verma, Kavita, P. Chatterjee, J. Shafi, J. Choi, and M. F. Ijaz, "A complete process of text classification system using state-of-the-art NLP models," Comput. Intell. Neurosci., vol. 2022, pp. 1–26, Jun. 2022.

[11] Y. Zhang, X. Yu, Z. Cui, S. Wu, Z. Wen, and L. Wang, "Every document owns its structure: Inductive text classification via graph neural networks," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020.

[12] X. Chen, P. Cong, and S. Lv, "A long-text classification method of Chinese news based on BERT and CNN," IEEE Access, vol. 10, pp. 34046–34057, 2022.

[13] J. Chen, Z. Yang, and D. Yang, "MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020.

[14] Z. Wang, P. Wang, L. Huang, X. Sun, and H. Wang, "Incorporating hierarchy into text encoder: A contrastive learning approach for hierarchical text classification," in Proc. 60th Annu. Meeting Assoc. Comput. Linguistics, 2022.

[15] K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: Hypergraph attention networks for inductive text classification," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020.

[16] Y. Lin, Y. Meng, X. Sun, Q. Han, K. Kuang, J. Li, and F. Wu, "BertGCN: Transductive text classification by combining GNN and BERT," in Proc. Findings Assoc. Comput. Linguistics (ACL-IJCNLP), 2021.

[17] Korapati, Dolasankar., Viswanath, G., G, Prathyusha. (2023). A Real-Time Video Based Vehicle Classification, Detection And Counting System. Industrial Engineering Journal, 52(9), 474–480.

[18] S. Hu, N. Ding, H. Wang, Z. Liu, J. Wang, J. Li, W. Wu, and M. Sun, "Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification," in Proc. Annu. Meeting Assoc. Comput. Linguistics, 2022.

[19] X. Sun, X. Li, J. Li, F. Wu, S. Guo, T. Zhang, and G. Wang, "Text classification via large language models," in Proc. Findings Assoc. Comput. Linguistics (EMNLP), 2023.

[20] Z. Wen and Y. Fang, "Augmenting low-resource text classification with graph-grounded pre-training and prompting," in Proc. 46th Int. ACM SIGIR Conf., Jul. 2023, pp. 506–516.

[21] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, "A diagnostic study of explainability techniques for text classification," in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), 2020.

[22] Viswanath, G., Abirami, V., & Prathyusha, G. (2024). Hybrid Feature Extraction With Machine Learning To Identify Network Attacks. International Journal Of HRM And Organizational Behavior, 12(3), 217–228.

[23] X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun, "PTR: Prompt tuning with rules for text classification," AI Open, vol. 3, pp. 182–192, Jan. 2022.

[24] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in Proc. 16th Conf. Eur. Chapter Assoc. Comput. Linguistics, 2021.

[25] A. Karimi, L. Rossi, and A. Prati, "AEDA: An easier data augmentation technique for text classification," in Proc. Findings Assoc. Comput. Linguistics: EMNLP, 2021.

[26] H. H. Luong, L. T. T. Le, and H. T. Nguyen, "An approach for web content classification with FastText," in Proc. Int. Conf. Comput. Data Social Netw., 2024, pp. 138–146.

[27] Q. Li, "Research on text sentiment classification and recognition based on improved BiLSTM and TextCNN," in Proc. 2nd Int. Conf. Mechatronics, IoT Ind. Informat., Jun. 2024, pp. 210–219.

[28] T. Pavlov and G. Mirceva, "COVID-19 fake news detection by using BERT and RoBERTa models," in Proc. 45th Jubilee Int. Conv. Inf., Commun. Electron. Technol. (MIPRO), May 2022, pp. 312–316.

[29] G, Viswanath., N, Madhvik., K, Bhaskar., K, Supriya. (2024). Machine-Learning-Based Cloud Intrusion Detection. International Journal of Mechanical Engineering Research and Technology, 16(9), 38–52.

[30] D. Madras, E. Creager, T. Pitassi, and R. S. Zemel, “Learning adversarially fair and transferable representations,” in Proc. 35th Int. Conf. Mach. Learn. (ICML), 2018, pp. 3381–3390.