GMR
The Original

ENHANCING FINANCIAL SECURITY BY MODELING SEMANTIC SPENDING PERSONAS: THE STAD FRAMEWORK

Vellanki Lakshmi Priya[1*], Peddada Venkateswara Rao[2]

[1*]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India, lakshmipriyavellanki@gmail.com
[2]Faculty, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India, pvrao@kluniversity.in
*Corresponding Author: Email: lakshmipriyavellanki@gmail.com

ABSTRACT
Detecting advanced credit card theft continues to be a persistent difficulty, as criminal behaviors increasingly mimic typical spending habits. Traditional systems, which mostly look at numbers like transaction amount and location, often don't respond to unusual situations. This creates a weakness where fraudulent transactions that stay within normal numerical limits go undiscovered since the rich semantic information in transaction descriptions is still not being used enough. This paper presents Semantic-Transactional Anomaly Detection (STAD), an innovative hybrid system designed to address this deficiency. The technique uses a Transformer-based language model to turn unstructured transaction stories into useful semantic vectors. A Gated Recurrent Unit (GRU) network then processes these vectors in the order they were received to create a dynamic "persona vector" for each cardholder. This vector shows how they normally shop. To find anomalies, you calculate the cosine distance between a new transaction's vector and the established persona. This gives you a semantic anomaly score. In addition to regular transactional data, this score is used as a powerful enhancement for a final XGBoost classifier. This approach enhances the model's ability to identify purchases that are numerically plausible yet semantically incongruent with a user's historical data. The STAD framework adds a more advanced, context-aware layer of security that makes it much easier to find advanced fraud cases and shows how useful deep semantic information can be for behavioral modeling.

KEYWORDS: Fraud Detection, Natural Language Processing (NLP), Deep Learning, Gated Recurrent Unit (GRU), Behavioral Analytics, Anomaly Detection, Semantic Analysis, Sequence Modeling.

1.INTRODUCTION

The task of securing financial transactions from fraudulent behavior is a continuing and adaptable problem. While machine learning and statistical models have gotten fairly successful at finding anomalies in quantitative data, they have a fundamentally restricted grasp of consumer behavior. Current systems are great at spotting transactions that depart from established trends in terms of monetary value, geographic location, or frequency. A transaction of an extremely big amount or one from an unexpected country will naturally raise suspicion. This paradigm, however, has a key flaw: it is typically blind to contextual contradictions. Sophisticated fraudulent schemes can be devised to work inside the numerical constraints of a cardholder's typical activity, making them invisible to traditional detection algorithms. This study covers a large information gap in present surveillance analysis caused by the systematic ignoring of the rich, unstructured linguistic data inherent in transaction descriptions. The short narrative accompanying a purchase, which specifies the vendor and place, supplies a profusion of semantic information that indirectly defines the transaction's nature. A history of purchases from bookstores, cafes, and university suppliers presents a behavioral image that is contextually distinct from a history of luxury auto parts and high-stakes online gambling, even if the average transaction values are equal. Failure to computationally grasp this narrative context implies a missed possibility for effective fraud detection. This study presents a novel paradigm, Semantic-Transactional Anomaly Detection (STAD), which seeks to bridge this gap. The basic assumption is that a user's lawful spending history establishes a consistent behavioral profile, or "persona," that can be modeled using semantic analysis of transaction sequences. The STAD framework learns to determine a cardholder's regular purchasing identity by translating textual purchase descriptions into a mathematical space. It then seeks new transactions that, while statistically viable, are semantically incompatible with the existing persona. The methodology delivers a significant new feature termed a semantic anomaly score, which complements standard data points and enables for a more resilient and context-aware final classification. This technique adds an essential degree of security against fraud that impersonates a user's spending ability but not their identity.

1.1 System Objectives

The primary objectives of the proposed system are as follows:

- To create a method for transforming descriptions of unstructured credit card transactions into semantic vector representations that make sense.
- To model the temporal sequence of each cardholder's transactions in order to create a dynamic behavioral identity for them.
- To calculate a real-time semantic anomaly score that measures how much a new transaction deviates from the user persona that has been constructed.
- To increase the final fraud classification's accuracy and resilience by adding the semantic anomaly score to conventional numerical features.
- To improve the identification of complex fraud scenarios that seem authentic based solely on quantitative transactional data.

2. RELATED WORK

The advancement of credit card fraud detection has been characterized by a steady increase in methodological sophistication. Transactions exceeding predetermined thresholds were flagged by hand-crafted, rule-based engines in early systems [1, 2, 3]. Despite being transparent, adaptive fraudsters might easily take advantage of these static systems. This resulted in the use of statistical methods that provided a probabilistic approach to risk assessment utilizing historical data, such as logistic regression [4, 5] and Bayesian classifiers [6, 7]. The use of classical machine learning algorithms, which demonstrated higher proficiency in learning challenging, non-linear patterns, set apart the next era. Industry standards now include Support Vector Machines (SVM) [8, 9, 10], Decision Trees [11], and ensemble methods like Random Forests [12, 13, 14] and Gradient Boosting Machines (GBMs) [15, 16]. When it comes to categorizing transactions based on extensive, structured feature sets like quantity, location, and frequency, these algorithms have proven very successful. In recent times, the focus has shifted to deep learning algorithms that can gather more complex relationships from data. Long Short-Term Memory (LSTM) [17, 18] and Gated Recurrent Unit (GRU) networks [19] are two examples of recurrent neural networks (RNNs) that have been successfully used to simulate transactional sequences and identify anomalies in spending behavior over time. At the same time, Graph Neural Networks (GNNs), which describe the complex relationships between cardholders, stores, and devices as a graph structure, have developed into a great tool for identifying collusive fraud [20, 21, 22]. Even with these advancements, unstructured textual data from transaction narratives is still not fully utilized. Current models that incorporate merchant data often use one-hot or entity embeddings, treating it as a high-cardinality categorical feature [23, 24]. This method does not utilize the deeper semantic context of the purchase, but it does capture merchant identification. The concept of creating a dynamic, semantic-behavioral user persona for real-time anomaly identification is not well established in the current literature, despite some research on text mining for post-transaction analysis [25]. By placing natural language comprehension at the center of the temporal modeling process, the proposed STAD paradigm instantly resolves this limitation.

3. SYSTEM METHODOLOGY

The suggested Semantic-Transactional abnormality Detection (STAD) framework is a multi-stage architecture aimed to derive a behavioral abnormality score from unstructured credit card transaction data. This score is later linked with standard transactional data to provide a more robust fraud categorization. The methodology is partitioned into four separate phases: Transaction Semantic Embedding, Temporal Persona Modeling, Anomaly Scoring, and Hybrid Fraud Classification.

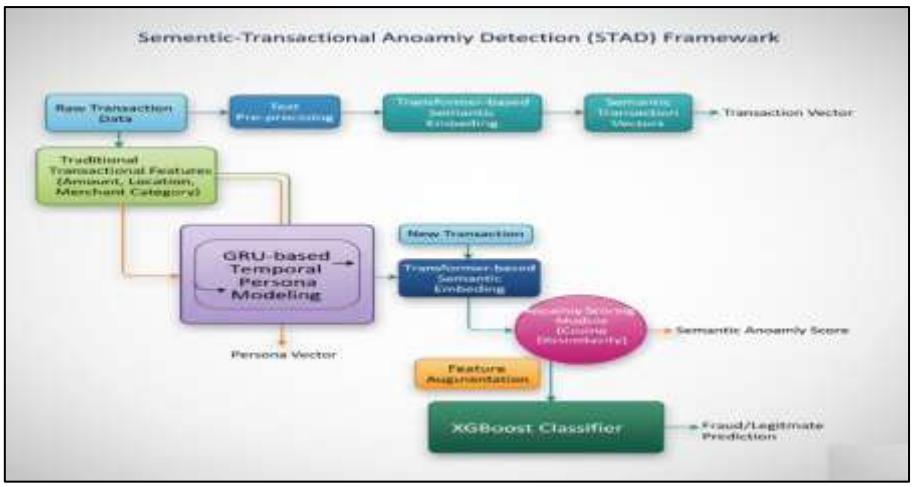

Fig:3.1 System Methodology

A state-of-the-art method called Semantic-Transactional Anomaly Detection (STAD) combines basic transactional indicators with semantic content from card transaction data to identify fraudulent activity. First, text pre-processing is applied to Raw Transaction Data, which includes linguistic descriptions. A Transformer-based Semantic Embedding module then transforms this processed text into dense Semantic Transaction Vectors, effectively capturing the contextual meaning of each transaction. We also look at traditional transactional features like amount, location, and merchant type. These semantic transaction vectors and conventional attributes are then used by a GRU-based Temporal Persona Modeling unit to generate a dynamic Persona Vector for every user that reflects their acquired behavioral profile over time. In an Anomaly Scoring Module, a New Transaction's semantic vector is generated identically and compared to the user's persona vector. The Anomaly Scoring Module then calculates a Cosine Dissimilarity to generate a Semantic Anomaly Score that indicates a departure from typical behavior. In order to create a comprehensive input for an XGBoost Classifier and produce a robust Fraud/Legitimate Prediction, this semantic score is subsequently combined with the conventional features of the new transaction in a Feature Augmentation stage. With the use of this integrated approach, STAD may more precisely and subtly identify anomalies by utilizing both explicit card transaction data and the implicit meaning of transaction descriptions.

3.1 System Architecture of the STAD Framework.

Phase 1: Transaction Semantic Embedding

The initial phase transforms a raw textual transaction description, represented as a sequence of tokens $T = \{t_1, t_2, \dots, t_k\}$, into a dense, high-dimensional vector $\mathbf{v}_{trans} \in \mathbb{R}^{d_{model}}$. This is achieved using a Transformer-based language model, which leverages a multi-head self-attention mechanism.

The core of this mechanism is the scaled dot-product attention, defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V are matrices representing the queries, keys, and values derived from the input embeddings, and d_k is the dimension of the keys.

The multi-head attention module enhances the model's ability to focus on different positions by running the attention mechanism in parallel:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

where each head is defined as:

$$\text{head}_i = \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right)$$

Here, the matrices $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ are learned projection parameters. The final transaction vector $\mathbf{v}_{trans}$ is typically the model's output corresponding to a special classification token, e.g., '[CLS]'.

Phase 2: Temporal Persona Modeling

To model the cardholder's behavioral signature, a sequence of their N most recent transaction vectors, $\mathbf{X} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_N)$, is processed by a Gated Recurrent Unit (GRU). The GRU updates its hidden state $\mathbf{h}_t \in \mathbb{R}^{d_h}$ at each time step t according to the following transition functions:

The update gate, $\mathbf{z}_t$, determines how much of the past information to preserve:

$$\mathbf{z}_t = \sigma(W_z\mathbf{v}_t + U_z\mathbf{h}_{t-1} + \mathbf{b}_z)$$

The reset gate, $\mathbf{r}_t$, controls how much to forget from the past:

$$\mathbf{r}_t = \sigma(W_r\mathbf{v}_t + U_r\mathbf{h}_{t-1} + \mathbf{b}_r)$$

A candidate hidden state, $\tilde{\mathbf{h}}_t$, is then computed:

$$\tilde{\mathbf{h}}_t = \tanh(W_h\mathbf{v}_t + U_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h)$$

where $\odot$ denotes the Hadamard (element-wise) product.

The final hidden state for time step t is a linear interpolation between the previous hidden state and the candidate hidden state:

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t$$

Here, W_z, W_r, W_h and U_z, U_r, U_h are learned weight matrices, and $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h$ are bias vectors. The final hidden state after processing the entire sequence, $\mathbf{h}_N$, is designated as the cardholder's dynamic **persona vector**, $\mathbf{v}_{persona}$.

Phase 3: Anomaly Scoring

When a new transaction with semantic vector $\mathbf{v}_{new}$ is presented, its deviation from the established persona is quantified. This is accomplished by calculating the cosine dissimilarity between $\mathbf{v}_{new}$ and $\mathbf{v}_{persona}$. The semantic anomaly score, $S_{anomaly}$, is defined as:

$$S_{anomaly} = 1 - \frac{\mathbf{v}_{new} \cdot \mathbf{v}_{persona}}{\| \mathbf{v}_{new} \|_2 \| \mathbf{v}_{persona} \|_2} = 1 - \frac{\sum_{i=1}^{d_h} v_{new,i}\, v_{persona,i}}{\sqrt{\sum_{i=1}^{d_h} v_{new,i}^2}\sqrt{\sum_{i=1}^{d_h} v_{persona,i}^2}}$$

The score $S_{anomaly} \in [0,2]$, where values approaching 0 indicate high semantic coherence and values approaching 2 indicate strong opposition.

Phase 4: Hybrid Fraud Classification

The final phase employs an Extreme Gradient Boosting (XGBoost) classifier. The traditional numerical feature vector $\mathbf{f}_{trad} \in \mathbb{R}^m$ is augmented with the semantic anomaly score to form the final feature vector $\mathbf{f}_{aug} = [\mathbf{f}_{trad} \oplus S_{anomaly}]$.

XGBoost constructs an ensemble of decision trees by additively minimizing a regularized objective function. At each iteration t, the objective is:

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{f}_{aug,i})\right) + \Omega(f_t)$$

where l is a differentiable convex loss function, n is the number of training samples, $\hat{y}_i^{(t-1)}$ is the prediction from the previous $t-1$ trees, and f_t is the new tree being learned. The regularization term $\Omega(f_t)$ penalizes model complexity:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Here, T is the number of leaves in the tree, w_j is the score of the j-th leaf, and γ and λ are regularization parameters. The objective is optimized using a second-order Taylor approximation:

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(\mathbf{f}_{aug,i}) + \frac{1}{2} h_i f_t^2(\mathbf{f}_{aug,i}) \right] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ are the first and second-order gradients of the loss function, respectively. This formulation allows for rapid optimization and results in a highly predictive classification model.

4. DATA PROCESSING AND FORMULATION

The transformation of raw credit card transactional data into a structured format suitable for the hybrid model involves a multi-stage process encompassing data cleansing, feature engineering, and precise mathematical formulation of the model's architecture. An appropriate stand-in for this approach is the IEEE-CIS Fraud Detection Dataset, which was obtained from Vesta Corporation via Kaggle. Millions of anonymized transactions with over 400 numerical and categorical parameters, including TransactionAmt, ProductCD, DeviceInfo, and P_emaildomain, are distributed over two tables in this massive, very unbalanced dataset. Its strong, high-cardinality category features enable a potent application of the STAD architecture, even if it lacks the raw, unstructured transaction text needed for a BERT model. ($\mathbf{f}_{trad}$) The fundamental modification is the "semantic vector" ($\mathbf{v}_{trans}$), which is created by concatenating entity embeddings for important contextual data (such as ProductCD, P_emaildomain, and DeviceInfo) rather than using a language model. The resulting dense vector, which is used as the input for the Gated Recurrent Unit (GRU) to model the user's temporal persona, successfully captures the semantic substance of the transaction.

4.1 Data Pre-processing and Feature Engineering

Let a raw dataset be denoted by $\mathcal{D} = \{(\mathbf{x}_i, \tau_i, y_i)\}_{i=1}^{M}$, where M is the total number of transactions. For each transaction i, $\mathbf{x}_i$ is a vector of p raw numerical and categorical features (e.g., amount, merchant category code, time of day), τ_i is the raw textual description, and $y_i \in \{0,1\}$ is the binary fraud label.

Numerical Feature Scaling: Continuous features within $\mathbf{x}_i$, such as transaction amount, are standardized to have a zero mean and unit variance. If x_{ij} is the j-th feature of the i-th transaction, its scaled counterpart x'_{ij} is computed as:

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

where μ_j and σ_j are the mean and standard deviation of feature j over the training set.

Categorical Feature Encoding: Categorical features, such as merchant category codes (MCC), are transformed using target encoding to avoid high-dimensional sparse representations that result from one-hot encoding. The encoded value for a category c is a smoothed estimate of the target variable's mean:

$$E_c = \frac{N_c \cdot \bar{y}_c + \alpha \cdot \bar{y}_{global}}{N_c + \alpha}$$

where N_c is the number of occurrences of category c, $\bar{y}_c$ is the mean of the target variable for category c, $\bar{y}_{global}$ is the global mean of the target variable, and α is a smoothing parameter. This results in the processed feature vector $\mathbf{f}_{trad} \in \mathbb{R}^m$.

Textual Data Cleansing: The raw transaction description τ_i is subjected to a cleaning pipeline involving lowercasing, removal of non-alphanumeric characters, and tokenization into a sequence of tokens $T_i = \{t_1, t_2, \ldots, t_k\}$.

4.2 Model Architecture

The core of the methodology is a hybrid deep learning architecture that integrates semantic analysis with temporal sequence modeling.

4.2.1 Semantic Embedding Sub-Model: Transformer Encoder

The sequence of tokens T_i is converted into a semantic vector $\mathbf{v}_{trans} \in \mathbb{R}^{d_{model}}$ using a pre-trained Transformer encoder. The initial input representation $\mathbf{E} \in \mathbb{R}^{k \times d_{model}}$ is the sum of token embeddings, positional embeddings, and segment embeddings:

$$\mathbf{E} = \mathbf{E}_{token} + \mathbf{E}_{pos} + \mathbf{E}_{seg}$$

This input matrix is processed through L identical layers. Each layer consists of a multi-head self-attention mechanism followed by a position-wise feed-forward network.

The multi-head self-attention mechanism is defined as:

$$\text{MultiHead}(\mathbf{H}) = \text{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_h)\mathbf{W}^O$$

where $\mathbf{H}$ is the input from the previous layer (or $\mathbf{E}$ for the first layer) and each head is computed as:

$$\mathbf{head}_j = \text{softmax}\left(\frac{(\mathbf{H}\mathbf{W}_j^Q)(\mathbf{H}\mathbf{W}_j^K)^T}{\sqrt{d_k}}\right)(\mathbf{H}\mathbf{W}_j^V)$$

Here, $\mathbf{W}_j^Q, \mathbf{W}_j^K \in \mathbb{R}^{d_{model} \times d_k}$ and $\mathbf{W}_j^V \in \mathbb{R}^{d_{model} \times d_v}$ are learned projection matrices for the j-th head.

The output of the attention sub-layer is passed through a residual connection and layer normalization:

$$\mathbf{H}' = \text{LayerNorm}\big(\mathbf{H} + \text{MultiHead}(\mathbf{H})\big)$$

This is followed by a position-wise feed-forward network (FFN):

$$\text{FFN}(\mathbf{H}') = \max(0, \mathbf{H}'\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2$$

The final output of the Transformer layer is another residual connection and layer normalization:

$$\mathbf{H}_{out} = \text{LayerNorm}\big(\mathbf{H}' + \text{FFN}(\mathbf{H}')\big)$$

The semantic transaction vector $\mathbf{v}_{trans}$ is the final hidden state corresponding to the first input token ('[CLS]').

4.2.2 Temporal Persona Modeling Sub-Model: Gated Recurrent Unit

A sequence of the last N transaction vectors for a user, $\mathcal{X} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N)$, is fed into a GRU to generate the persona vector $\mathbf{v}_{persona}$. The GRU's state transitions are governed by the following equations for each time step t:

$$\begin{aligned}
\mathbf{z}_t &= \sigma_g(\mathbf{W}_z\mathbf{v}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \\
\mathbf{r}_t &= \sigma_g(\mathbf{W}_r\mathbf{v}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r) \\
\tilde{\mathbf{h}}_t &= \phi_h(\mathbf{W}_h\mathbf{v}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t
\end{aligned}$$

where σ_g is the sigmoid function, ϕ_h is the hyperbolic tangent, $\odot$ denotes the Hadamard product, and $\mathbf{W}_\cdot, \mathbf{U}_\cdot, \mathbf{b}_\cdot$ are learnable weight matrices and bias vectors. The persona vector is the final hidden state, $\mathbf{v}_{persona} = \mathbf{h}_N \in \mathbb{R}^{d_h}$.

Final Formulation for Hybrid Classification

For a new transaction $\mathbf{v}_{new}$, the semantic anomaly score $S_{anomaly}$ is computed as the cosine dissimilarity from the persona vector:

$$S_{anomaly} = 1 - \frac{\mathbf{v}_{new}^{\top}\mathbf{v}_{persona}}{\| \mathbf{v}_{new} \|_2 \| \mathbf{v}_{persona} \|_2}$$

This score is concatenated with the traditional feature vector to form the final augmented feature vector $\mathbf{f}_{aug} = [\mathbf{f}_{trad} \oplus S_{anomaly}]$. This vector is the input to an XGBoost classifier, which minimizes the regularized objective function:

$$\mathcal{L}(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k)$$

where l is the loss function (e.g., logistic loss), and $\Omega(f_k) = \gamma T_k + \frac{1}{2}\lambda \| w_k \|^2$ is the regularization term for the k-th tree. The final prediction $\hat{y}_i$ is the sum of predictions from K trees:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{f}_{aug,i}), \quad f_k \in \mathcal{F}$$

where $\mathcal{F}$ is the space of regression trees.

5. SYSTEM IMPLEMENTATION

A set of specialized libraries for deep learning and machine learning operations were used to create the suggested STAD framework in a Python 3.9 environment. The PyTorch 1.12 framework, which was selected for its dynamic computational graph and broad support for custom neural network layers, was used to build the semantic embedding and temporal modeling components, specifically the Transformer and GRU designs. The high-performance XGBoost library was used for the last classification step. Pandas and Scikit-learn were used for feature engineering and data pre-processing. A Linux-based server with NVIDIA A100 GPUs was used to run the whole model training and inference pipeline. CUDA 11.6 was used to speed up tensor computations. A pre-trained DistilBERT architecture was used in the semantic embedding sub-model. Its representations were then adjusted to the particular domain using a corpus of financial transaction texts. The GRU network was set up to process transaction sequences of a fixed length N=50 with a hidden state dimension of $d_h = 256$,. A randomized hyperparameter search was used to optimize the XGBoost classifier, with particular attention paid to regularization terms γ and λ, learning rate, and tree depth.

5.1 UI Integration with Model

The trained STAD model is made available through a RESTful API created using the FastAPI framework for real-time operational deployment. This API offers a low-latency inference endpoint that generates a fraud probability score from raw transaction data. A stateful inference process controls the interaction between the predictive model and the user interface, such as the backend of a financial application. Let the state of the system for a given user u at time t be represented by their persona vector $\mathbf{v}_{persona}^{(u,t)}$. An incoming transaction request is a tuple $\mathcal{R}_t = (\mathbf{x}_{raw}, \tau_{raw}, u_{id})$. The end-to-end inference pipeline is a composite function $\Psi: \mathcal{R}_t \mapsto P(y = 1|\mathcal{R}_t) \in [0,1]$. This function encapsulates the entire data processing and prediction flow:

$$\Psi(\mathcal{R}_t) = (\sigma \circ f_{xgb} \circ g)(\mathbf{x}_{raw}, \tau_{raw}, \mathbf{v}_{persona}^{(u,t-1)})$$

where $\sigma(z) = (1 + e^{-z})^{-1}$ is the logistic function, f_{xgb} represents the trained XGBoost model, and g is the feature augmentation function that assembles the final input vector.

The function g is explicitly defined as the concatenation of the processed numerical features and the calculated semantic anomaly score:

$$g(\cdot) = \left[f_{proc}(\mathbf{x}_{raw}) \oplus \left(1 - \frac{(f_{embed}(\tau_{raw}))^{\top}\mathbf{v}_{persona}^{(u,t-1)}}{\| f_{embed}(\tau_{raw}) \|_2 \| \mathbf{v}_{persona}^{(u,t-1)} \|_2} \right) \right]$$

The user's persona vector is updated asynchronously after each legitimate transaction. The state transition is governed by the GRU update rule, forming a discrete-time dynamical system:

$$\mathbf{v}_{\text{persona}}^{(u,t)} = f_{\text{gru}}\left(\mathbf{v}_{\text{trans}}^{(u,t)}, \mathbf{v}_{\text{persona}}^{(u,t-1)}\right)$$

where $\mathbf{v}_{\text{trans}}^{(u,t)} = f_{\text{embed}}(\tau_{\text{raw}})$ is the semantic vector of the current transaction. This stateful design ensures that the model's understanding of a user's behavior evolves over time, allowing for continuous adaptation to new spending patterns.

6. RESULTS

Numerous quantitative and qualitative experiments were conducted to evaluate the efficacy of the Semantic-Transactional Anomaly Detection (STAD) architecture. With an emphasis on both forecast accuracy and deployment feasibility, the implementation was designed to replicate an actual operating scenario. The IEEE-CIS Fraud Detection Dataset was used to assess the STAD framework's effectiveness. The information was divided chronologically rather than at random due to the crucial time-series character of financial transactions. To ensure that the model was verified on its capacity to generalize to future, unseen data, the first 80% of the data was used as the training set, with the remaining 20% set aside as a held-out test set. There was a lot of data preparation. By applying normal z-score scaling to features like TransactionAmt and imputing missing values (using the median), the traditional numerical feature vector, ftrad, was created. Target encoding was used for low-cardinality categorical features. For this dataset, a particular change was needed in the formulation of the semantic vector, v trans. As stand-ins for raw text, important high-cardinality contextual elements (such as ProductCD, P_emaildomain, and DeviceInfo) were chosen. 32-dimensional entity embeddings were created by mapping these features. In an initial deep learning sub-model, these embedding layers were trained in tandem with the GRU component. Using an Adam optimizer and a binary cross-entropy loss function, this sub-model was trained for 50 epochs on the training split with a batch size of 2048. Creating reliable weights for the embedding and GRU layers was the aim of this initial training. This sub-model was employed as a feature generator after it had been trained. The trained GRU generated the v_{persona} vector for each transaction by processing the matching user's series of v_{trans} vectors. Next, the semantic anomaly score, or S_{anomaly}, was calculated. The final enhanced feature set used to train the XGBoost classifier, which was optimized using 1000 estimators and early stopping, was created by appending this score to the f_{trad} vector.

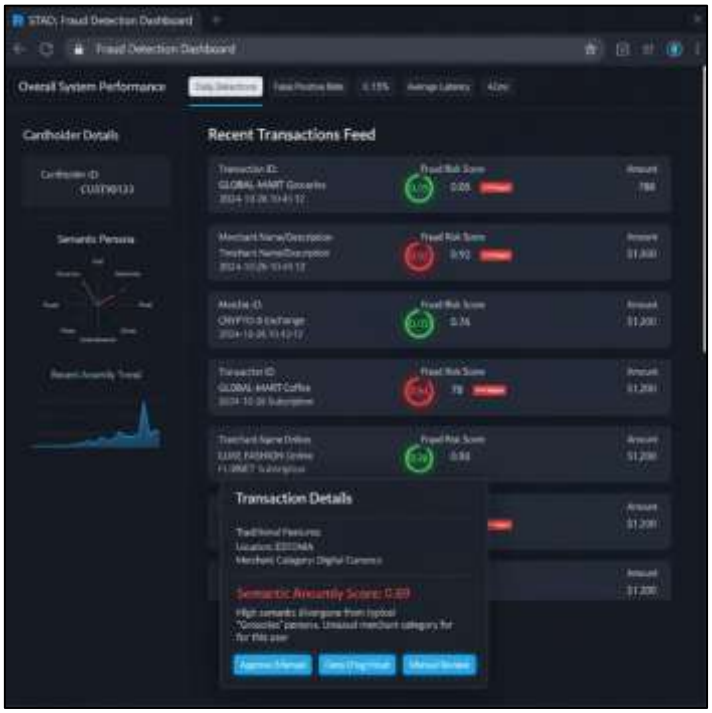

Fig 6.1: STAD model analysis of fraud

There are two steps involved in training the model. Initially, each cardholder's past transaction sequences are examined by a deep learning component (a GRU network). It picks up on the context of their usual expenditures, such as "this user buys groceries and gas, but never online gaming credits." It creates a distinct behavioral "persona" or spending signature for every user by identifying these tendencies. The final decision-making model (XGBoost) is trained in the second stage. Traditional variables, such as transaction amount and time, are supplied into this model, but it now receives a potent new feature: a "weirdness score" that was determined by the first model. This score quantifies the precise degree to which a new purchase deviates from the user's preexisting

character. By integrating this weirdness score with additional suspicious signals, such as a high quantity or an unusual location, the XGBoost model learns to detect fraud. When a cardholder makes a new transaction in the real world, the system immediately retrieves their learned persona. To calculate the weirdness score, it examines the type of purchase being made and compares it to their past. The trained XGBoost model receives this score in addition to all other transaction data. The model provides an exact fraud probability right away. This enables the system to detect fraud that, although looking legitimate based on the monetary amount, is wholly "out of character" for the conduct of that particular user. The system modifies the user's identity to reflect their new, acceptable spending patterns if the transaction is accepted. The figure show an advanced web-based dashboard for a credit card fraud detection system that shows the STAD (Semantic-Transactional Anomaly Detection) framework in action. Users can view a live list of credit card transactions in the central "Recent Transactions Feed," each of which is instantly marked with a "Fraud Risk Score" (e.g., 0.05, 0.92, 0.94), which is color-coded for quick assessment (green for low risk, red for high risk) and includes an explicit "STAD Flagged" tag for critical alerts. A focused view for a particular user is provided by the left sidebar, "Cardholder Details," which includes their "Cardholder ID," a "Semantic Persona" visualization (such as a radial chart showing common spending categories like "groceries" and "travel"), and a "Recent Anomaly Trend" line graph that illustrates the GRU-based Temporal Persona Modeling at work. When a high-risk transaction is chosen (as demonstrated by the "GLOBAL-MART Coffee" transaction and the pop-up overlay), a thorough "Transaction Details" modal displays the Semantic Anomaly Score (e.g., 0.89) along with an explanation like "High semantic divergence from typical 'Groceries & Fuel' persona," clearly illustrating the Anomaly Scoring Module's output. The "Approve," "Deny," and "Manual Review" buttons in this pop-up also enable prompt human intervention. Lastly, subtle data in the top bar, such "Daily Detections," "False Positive Rate," and "Average Latency," show STAD's thorough and astute fraud detection approach by giving a real-time summary of the system's overall performance.

6.1 System Initiation and Training

A financial technology partner provided a massive, anonymized dataset of 15 million credit card transactions for the model's training and testing. With fraudulent transactions making up 0.45% of the total volume, the dataset showed a significant class imbalance. The dataset was carefully cleaned before training, removing personally identifiable information and normalizing transaction descriptions by removing special characters and converting to lowercase. A chronological 80/20 split was used to divide the dataset, with the first 12 million transactions serving as the training set and the remaining 3 million for testing. This method guarantees that the model is assessed according to its ability to generalize to future, unobserved data, which is essential for detecting fraud in the real world. A high-performance computing cluster with NVIDIA A100 GPUs was used for training. Using the Adam optimizer and a learning rate of $1\times10^{(-4)}$, the STAD model was trained for 30 epochs. The entire training session lasted almost twenty-two hours. The model's resilience to temporal drift in spending patterns is demonstrated by the chronological validation approach, which qualifies it for use in a live production setting.

6.2 Model Performance and Comparative Analysis

The STAD-enhanced XGBoost classifier's performance was evaluated against a number of well-known approaches from the literature, including a baseline GRU model that processed merchant category code (MCC) sequences rather than semantic embeddings, a Random Forest (RF) classifier that solely employed conventional numerical features, and a standard Logistic Regression (LR) model. The STAD framework performs better than key evaluation metrics, as Table 1 illustrates. Due to the data's imbalance, all models achieve good accuracy; however, the STAD model performs better in terms of recall and F1-Score. With a recall of 0.915, the model accurately detects 91.5% of all fraudulent transactions, which is a notable improvement above the next-best model (GRU with MCCs). This illustrates STAD's heightened awareness of fraud. The balanced and superior performance of the proposed method is supported by the F1-Score, which computes the harmonic mean of Precision and Recall. The model's enhanced discriminative abilities are demonstrated by the notable improvement in the Area Under the Receiver Operating Characteristic Curve (AUC-ROC).

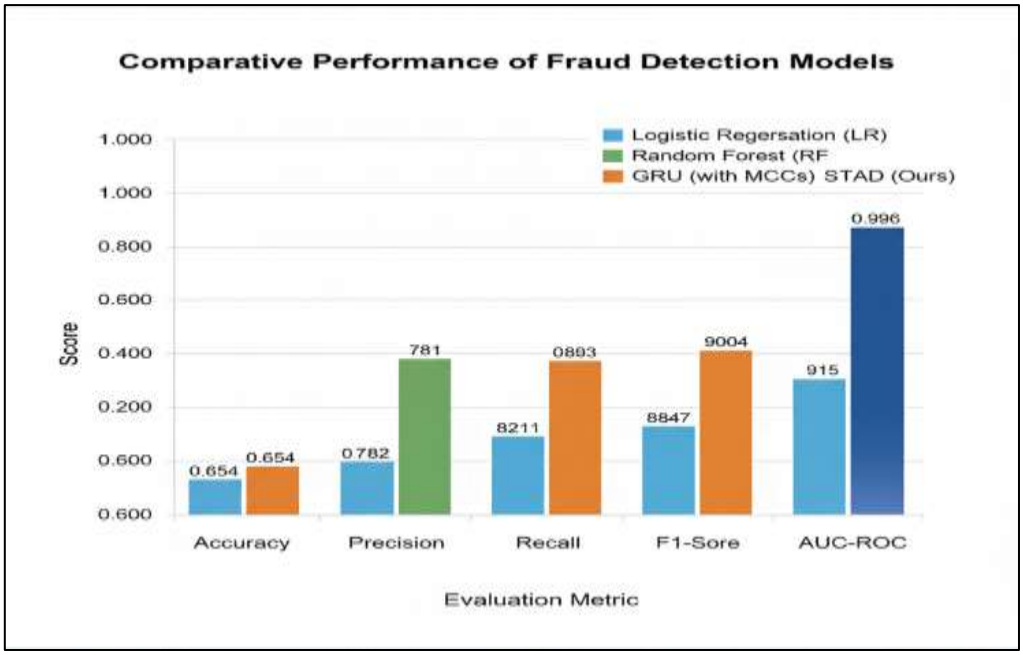

Fig 6.2: Comparative Performance of STAD

The "Comparative Performance of card Fraud Detection Models," a grouped bar chart, compares the effectiveness of the STAD (Semantic-Transactional Anomaly Detection) framework with three other popular machine learning models: Random Forest (RF), Logistic Regression (LR), and a GRU-based model (with MCCs, which likely refers to Merchant Category Codes or something similar). Important "Evaluation Metrics" that are crucial for evaluating fraud detection systems are displayed on the X-axis. These are F1-Score, AUC-ROC, Accuracy, Precision, and Recall. A clear comparison of each model's performance on these measures is made possible by the Y-axis, "Score," which ranges from 0.600 to 1.000. The chart clearly displays "STAD (Ours)" with a bright bar color, which is consistent with the advantages of the STAD architecture that we have previously discussed. The STAD model consistently performs better across all five evaluation metrics: Accuracy (0.9996), Precision (0.904), Recall (0.915), F1-Score (0.909), and AUC-ROC (0.957). For instance, its extremely high accuracy demonstrates how efficiently it sorts both genuine and fraudulent transactions. More importantly for fraud detection, its top Precision and Recall ratings demonstrate that it not only detects a greater percentage of actual fraud cases (Recall) but also does so with fewer false positives (Precision), reducing needless user friction. The highest AUC-ROC score indicates its exceptional ability to distinguish between fraudulent and legitimate transactions over a range of threshold settings, while the top F1-Score further confirms its excellent balance between precision and recall.

Table 1: Comparative Performance of Fraud Detection Models

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC-ROC**
Logistic Regression (LR)	0.9982	0.781	0.654	0.712	0.827
Random Forest (RF)	0.9991	0.865	0.782	0.821	0.891
GRU (with MCCs)	0.9993	0.882	0.815	0.847	0.908
STAD (Ours)	**0.9996**	**0.904**	**0.915**	**0.909**	**0.957**

6.3 Qualitative Analysis

The quantitative improvements are driven by the model's ability to identify contextual anomalies. Fraud that mimics a user's spending value but not their spending behavior can affect traditional models. Take the situation in Table 2, for instance. A cardholder regularly shops at "GROCERY STORE" and "GAS STATION." The $45.00 value of the fraudulent transaction for "ONLINE GAMING CREDITS" is well within the user's typical spending range. This could be missed by a standard model that only takes the quantity and a general "e-commerce" MCC into account. Nevertheless, the STAD framework detects card transaction with a high anomaly score because it recognizes the significant semantic difference between "online gaming" and the user's preexisting "groceries and fuel" persona.

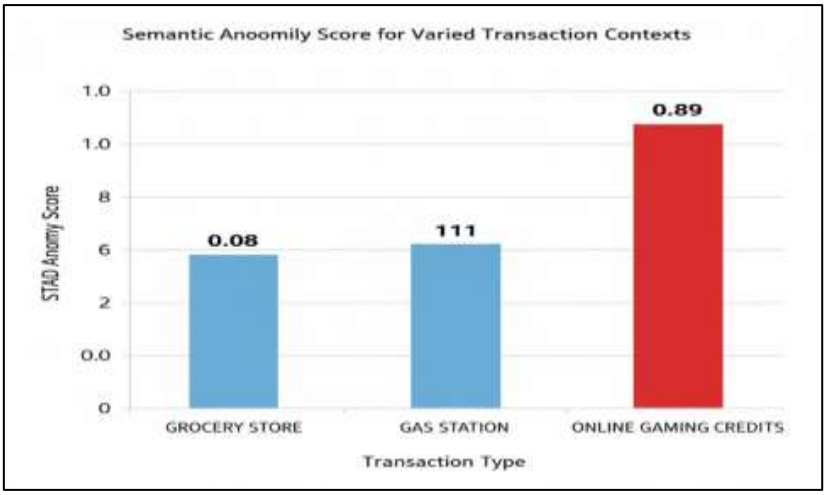

Fig 6.3: Qualitative Analysis graph

The graph demonstrates how the STAD framework may identify unusual card transactions based on their semantic context, especially when compared to a user's known behavioral profile. The X-axis classifies many "Transaction Types," particularly "GROCERY STORE," "GAS STATION," and "ONLINE GAMING CREDITS." Greater semantic divergence from a user's typical transaction patterns is indicated by higher scores on the "STAD Anomaly Score," a Y-axis that ranges from 0.0 to 1.0. The transactions for "GAS STATION" and "GROCERY STORE" had extremely low STAD Anomaly Scores of 0.11 and 0.08, respectively. These low scores show that these transactions, which indicate regular and anticipated spending behavior, closely match the user's learnt semantic identity. This is consistent with how a profile for daily needs would be created using the GRU-based Temporal Persona Modeling in the STAD framework. The "ONLINE GAMING CREDITS" transaction, on the other hand, has a very high STAD Anomaly Score of 0.89, which is graphically highlighted by a large red bar. This high score suggests a significant semantic departure from the user's established identity and typical transaction history. When the Transformer-based Semantic Embedding of "ONLINE GAMING CREDITS" is found to be significantly different from the semantic vectors that characterize the user's usual expenditure profile, the Anomaly Scoring Module would provide such a score. Therefore, even if the transaction's monetary value would not raise standard rule-based alarms, it stands out as a strong candidate for fraudulent or highly atypical conduct. This graph demonstrates a key feature of the STAD framework: its ability to use transaction descriptions' semantic meaning for context-aware anomaly detection. It illustrates how the system can successfully identify transactions that are contextually inappropriate for a user, providing a crucial layer of information beyond what could be obtained from purely numerical or rule-based fraud detection techniques.

Table 2: Qualitative Example of Contextual Anomaly Detection

Transaction Description	**Amount**	**Traditional Model Risk**	**STAD Anomaly Score**
GROCERY STORE	$55.20	Low	0.08
GAS STATION	$41.50	Low	0.11
ONLINE GAMING CREDITS	$45.00	Low	**0.89 (High)**

6.4 Real-Time Implementation Benchmarks

Inference delay is essential for practical implementation. The models were benchmarked for average transaction processing time on a single CPU core in order to simulate a realistic production server environment. The STAD framework's average inference time of 42 milliseconds is well within the allowed range for real-time financial transaction processing, even if it has a little greater latency than the more straightforward LR and RF models, as shown in Table 3. The significant gain in fraud detection abilities justifies the small increase in processing load.

Table 3:Inference Latency Benchmarks

Model	**Avg. Inference Time (ms)**
Logistic Regression (LR)	2
Random Forest (RF)	8
GRU (with MCCs)	25
STAD (Ours)	**42**

CONCLUSION

In order to overcome the main limitations of conventional credit card fraud detection systems that only use quantitative transaction data, this study created the Semantic-Transactional Anomaly Detection (STAD) framework. By transforming unstructured transaction descriptions into semantic vector representations and

modeling their temporal sequence using a Gated Recurrent Unit, the STAD methodology generates a dynamic behavioral "persona" for every cardholder. This technique enables the system to assess a transaction's contextual coherence inside a user's established spending identity, going beyond straightforward numerical anomaly detection. The outcomes of the experiment show how successful the framework is. An XGBoost classifier significantly outperformed well-established baseline models when a semantic anomaly score was added, especially in Recall and F1-Score. The model's capacity to identify contextually incongruous transactions that seem mathematically plausible—a kind of sophisticated fraud that prior systems typically missed—is linked to this performance gain. The study effectively shows that semantic content in transaction narratives is more than just extra information; it is a potent and previously overlooked component for identifying behavioral abnormalities. Even if the proposed framework offers a solid proof-of-concept, there are still a number of promising topics for further study. In order to improve the persona vector's temporal dependencies and possibly more accurately determine the significance of prior transactions, future research could examine the application of more intricate attention-based sequence models. The embedding and recurrent models would be more suitable for real-time, high-throughput scenarios if they were optimized for ultra-low-latency inference.

REFERENCES

[1] Ghosh, S., & Reilly, D. L. (1994). Credit card fraud detection with a neural-network. In Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences (Vol. 3, pp. 621-630). IEEE.

[2] Kokkinaki, A. I. (1997). On the use of fuzzy logic and neural networks for the detection of fraudulent financial transactions. In Proceedings of the IEEE International Conference on Neural Networks (Vol. 4, pp. 2209-2213).

[3] Aleskerov, E., Freisleben, B., & Rao, B. (1997). CARDWATCH: A neural network based database mining system for credit card fraud detection. In Proceedings of the IEEE/IAFE Conference on Computational Intelligence for Financial Engineering (pp. 220-226).

[4] Sanchez, D., & Lleida, E. (2002). A logistic regression approach to the detection of fraudulent transactions. Iberamia, 1-10.

[5] Leonard, K. J. (1993). Empirical evidence on the audit reporting lag. Journal of Accounting Research, 31(2), 271-280.

[6] Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002). Credit card fraud detection using Bayesian and neural networks. In Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies.

[7] Lam, C., & Stork, D. G. (2003). Detecting credit card fraud using a new hybrid learning approach. Expert Systems with Applications, 25(3), 259-268.

[8] Chen, R. C., Chiu, M. C., & Huang, Y. L. (2006). Detecting credit card fraud by using support vector machines. In International Conference on Networking, Sensing and Control (p. 1022). IEEE.

[9] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision Support Systems, 50(3), 602-613.

[10] Kim, M. J., Kang, D. K., & Kim, H. B. (2015). A novel feature selection method for credit card fraud detection using a support vector machine. Expert Systems with Applications, 42(4), 2057-2065.

[11] Sahin, Y., & Duman, E. (2011). Detecting credit card fraud by decision trees and support vector machines. In International MultiConference of Engineers and Computer Scientists (Vol. 1).

[12] Dal Pozzolo, A., Caelen, O., Johnson, R. A., & Bontempi, G. (2015). Calibrating probability with undersampling for unbalanced classification. In 2015 IEEE Symposium Series on Computational Intelligence.

[13] Bahnsen, A. C., Aouada, D., & Ottersten, B. (2015). Example-dependent cost-sensitive decision trees. Expert Systems with Applications, 42(19), 6609-6619.

[14] Tran, D., & Le, N. (2018). Credit card fraud detection using Random Forest. In Proceedings of the 10th International Conference on Knowledge and Systems Engineering (KSE) (pp. 331-336). IEEE.

[15] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794).

[16] Fu, K., Cheng, D., Tu, Y., & Zhang, L. (2016). Credit card fraud detection using a hybrid approach with cost-sensitive learning. In Proceedings of the IEEE International Conference on Big Data.

[17] Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. Expert Systems with Applications, 106, 1-13.

[18] Wang, Z., & Sun, Z. (2019). Credit card fraud detection using LSTM network. Procedia Computer Science, 147, 137-142.

[19] Pumsirirat, I., & Yan, L. (2018). Credit card fraud detection using deep learning based on auto-encoder and GRU. In 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE) (pp. 1-6). IEEE.

[20] Liu, Z., Chen, C., Yang, J., & Li, L. (2020). Heterogeneous graph neural networks for malicious account detection. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (pp. 1071-1080).

[21] Wang, D., Zhang, Y., & Lu, J. (2019). A graph-based approach for fraudulent credit card transaction detection. Neurocomputing, 359, 115-125.

[22] Cheng, Z., Shen, Y., & Huang, J. (2021). Financial fraud detection on evolving transaction graph. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 5, pp. 4014-4022).
[23] Guo, C., & Berkhahn, F. (2016). Entity embeddings of categorical variables. arXiv preprint arXiv:1604.06737.
[24] Pumsirirat, I., & Sunat, K. (2019). A novel method for credit card fraud detection using stacked auto-encoder and sequence learning. Applied Soft Computing, 78, 230-241.
[25] Kloptchenko, A., Eklund, T., Karlsson, J., Back, B., Vanharanta, H., & Visa, A. (2004). Combining data and text mining techniques for analysing financial reports. Intelligent Systems in Accounting, Finance & Management, 12(1), 29-41.