An Explorative Guide on How to Detect Forged Car Insurance Claims with Language Models

Quentin Telnoff[1,2][a], Emanuela Boros[1][b], Mickael Coustaty[1][c], Fabrice Crohas[2],
Antoine Doucet[1][d] and Frédéric Le Bars[2]

[1] *University of La Rochelle, L3i, F-17000, La Rochelle, France*

[2] *Itesoft, F-30470, Aimargues, France*

Keywords: Forgery Detection, Tabular Data, Language Models.

Abstract: Detecting forgeries in insurance car claims is a complex task that requires detecting fraudulent or overstated claims related to property damage or personal injuries after a car accident. Building predictive models for detecting them raises several issues (e.g. imbalance, concept drift) that cannot only depend on the frequency or timing of the reported incidents. The difficulty of tackling this type of task is further intensified by the static tabular data generally used in this domain, while submitted insurance claims largely consist of textual data. We, thus, propose an explorative guide for detecting forged car insurance claims with language models. Specifically, we investigate two transformer-based frameworks: supervised (where the model is trained to differentiate between forged and non-forged cases) and self-supervised (where the model captures the standard attributes of non-forged claims). For handling static tabular data and unstructured text fields, we inspect various forms of data row modelling (table serialization techniques), different losses, and two language models (one general and one domain-specific). Our work highlights the challenges and limitations of existing frameworks.

1 INTRODUCTION

Financial fraud is gaining improper advantages or financial benefits by using illegal and fraudulent methods (Abdallah et al., 2016). This type of fraud can be committed in different areas, such as insurance, banking, taxation, and corporate sectors (Kirlidog and Asuk, 2012; Peng et al., 2006). Specifically, insurance fraud is a common phenomenon committed against insurance companies. According to the Insurance Fraud Bureau of Australia (IFBA), the cost of fraudulent claims incurred by the industry is more than $2 billion annually and represents 10% of reported claims (Itri et al., 2019a; Subudhi and Panigrahi, 2020). The rapid advancement of digital processes, which became primarily adopted with the COVID-19 pandemic, offered great potential for forgeries (insurance professionals believed that 20% of claims could contain fraud[1]).

When detecting forged car insurance claims, three main challenges stand out: the **data imbalance**, the **concept drift**, and the **tabular format of the data**.

First, the number of fraudulent financial transactions is far fewer than non-fraudulent ones, and this problem of imbalanced data distribution across classes generally affects the efficiency of machine learning models (Abdallah et al., 2016; Tennyson and Salsas-Forn, 2002).

Second, the fraud types change over time, and the effectiveness of these methods may diminish due to concept drift, necessitating frequent model retraining and rebalancing, which can be challenging in real-world situations (Ryman-Tubb et al., 2018).

Finally, when a claim is submitted to an insurance company, the information is converted into a tabular format to align with the structure of information systems (Ali et al., 2022). This format, commonly found in public datasets, is not readily conducive to processing and analysing the full extent of valuable information. These databases are composed of highly heterogeneous data, especially when it comes to unstructured free text, categorical and numerical data (Borisov et al., 2023). This limitation can hinder the effectiveness of traditional machine-learning approaches to fraud detection.

These approaches require text or categorical encoding in order to use mathematical models. However, these encodings lose information from the original

[a] https://orcid.org/0009-0009-1364-6242

[b] https://orcid.org/0000-0001-6299-9452

[c] https://orcid.org/0000-0002-0123-439X

[d] https://orcid.org/0000-0001-6160-3356

[1] https://www.friss.com/insight/insurance-fraud-report-2022/

Telnoff, Q., Boros, E., Coustaty, M., Crohas, F., Doucet, A. and Bars, F.
An Explorative Guide on How to Detect Forged Car Insurance Claims with Language Models.
DOI: 10.5220/0012232900003598
In *Proceedings of the 15th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2023) - Volume 1: KDIR*, pages 403-412
ISBN: 978-989-758-671-2; ISSN: 2184-3228

data (e.g. one-hot encoding (Jiao and Zhang, 2021), all texts and categories are at the same distance from each other, there is, therefore, a loss of textual semantic information and suffers from the curse of dimensionality).

In this explorative study, we systematically address the detection of forged car insurance claims. Our work involves:

- The **investigation of different transformations of heterogeneous tabular data** into a sentence in order to standardize content and use a large language model to handle the tabular format of the data;
- The exploration of **a self-supervised and a supervised framework** based on BERT, both in general and domain-specific variations;
- The use of **different types of loss functions** and the design of optimized threshold to tackle data imbalance.

The rest of the article is organized as follows. Section 2 reviews the related work about car insurance forgery detection, with tabular data modelling using supervised and unsupervised techniques. Section 3 describes the methodology used in the paper. Section 4 describes the experimental setup, i.e. describes the studied data with the preprocessing step, the metrics, and the parameters used in the experiments. Section 5 provides the description of the experiments, the results, and the analysis. Section 6 concludes this study with our main findings.

2 RELATED WORK

Fraud detection challenges, such as concept drift, skewed distribution, and data imbalance, were generally approached with fraud detection systems based on supervised approaches (Abdallah et al., 2016; Ali et al., 2022; Ryman-Tubb et al., 2018), such as support vector machines (SVMs) (Kirlidog and Asuk, 2012), XGboost and light gradient-boosting machine (LGBM) (Kate et al., 2023; Majhi et al., 2019), rule-induction techniques, decision trees, logistic regression, and meta-heuristics (e.g. genetic algorithms) (Ali et al., 2022; Sithic and Balasubramanian, 2013).

With regard to the car insurance claims datasets utilized in this study, several approaches were proposed that are in line with previous research (Abdallah et al., 2016; Tennyson and Salsas-Forn, 2002) i.e. data imbalance. This challenge has generally been tackled with supervised learning and data rebalancing techniques such as upsampling and oversampling methods (Gupta et al., 2021; Hassan and Abraham, 2016; Aslam et al., 2022), synthetic minority oversampling techniques (SMOTE) (Soufiane et al., 2022; Kate et al., 2023), undersampling approaches (e.g. fuzzy c-means clustering) (Subudhi and Panigrahi, 2020; Majhi et al., 2019; Nian et al., 2016). These methods not only demonstrate the significance of eliminating noisy and redundant samples from the majority class of highly skewed imbalanced datasets but also prove efficiency in terms of lowered false alarms while simultaneously controlling the imbalanced class distribution and systematic identification of fraudulent cases (Sundarkumar and Ravi, 2015).

In regard to concept drift, when a fraud detection system is set up, the models cannot be static, as the environment will evolve because fraud types will vary over time (Gama et al., 2014). Several methods were proposed to counter this phenomenon, such as re-training the model when the drift of a concept is detected, followed by removing minor relevant examples (Dal Pozzolo et al., 2015) or modelling the distribution of non-fraudulent data, which is likely to vary less in time (Krawczyk and Woźniak, 2015). Finally, clustering techniques can detect suspicious healthcare frauds from large databases (Peng et al., 2006).

Other research proposed deep learning (DL) models to gain pragmatic insights into the behaviour of an insured person using unsupervised variable importance. For example, variational autoencoders were trained to reconstruct non-fraud cases with minimal reconstruction error, with impressive results (Gomes et al., 2021). Finally, similar to our study, LogBERT is another self-supervised approach to anomaly detection in logs based on BERT with the objective of detecting anomalies in logs generated by online systems by learning the underlying patterns of normal log sequences and detecting if there are any deviations from these normal log patterns (Guo et al., 2021).

3 METHODOLOGY

We explore tabular data modelling with a transformer-based language model, which is decomposed into three steps presented in the workflow overview in Figure 1.

3.1 Table Serialization

Table serialization is a method for representing 2-dimensional tabular data into a 1-dimensional sequence of tokens suitable and understandable for a transformer-based model (Badaro and Papotti, 2022). It involves converting the rows and fields of a table into a linear sequence of tokens, such as words or subwords. This allows the model to learn the structure and

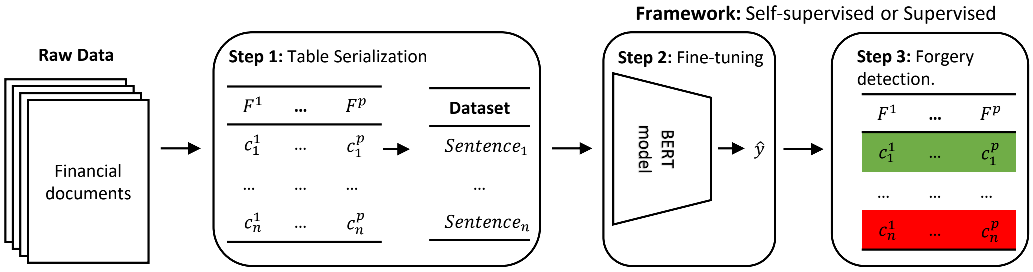


Figure 1: Workflow overview. It begins with information extraction from financial documents in tabular format (not covered in the article). Each row in the table represents a document. First, a transformation is applied to the rows of a table composed of fields in order to create an understandable and suitable input for a large language model (Step 1). Finally, in the two last steps, we fine-tune the BERT-based models (Step 2) in order to detect forged cases from authentic cases (Step 3).

relationships between the various elements of the table. More formally, let n and p be two non-zero integers, and let us consider a given table with n rows and p fields. The field names are noted $\{F^j\}_{j=1}^{p}$. Let $i \leq n$ and $j \leq n$, we define c_i^j, the cell of coordinates i and j, as the intersection of the i^{th} row and the j^{th} field. Depending on the tokeniser, a cell can be composed of one word, multiple words, or multiple sub-words. We, thus, propose three table serialization transforms.

Cell Concatenation Transform (CC). The first transform consists of the concatenation of the tokens separated by tokenizer-specific special markers $[CLS]$ and $[SEP]$ (Eq. 1).

$$CC_i = \text{“}[CLS]\ c_i^1\ [SEP]\ c_i^2\ [SEP]\ ...\ [SEP]\ c_i^p\ [SEP]\text{''} \quad (1)$$

Field & Cell Concatenation Transform (FCC). In the second transformation (Eq. 2), the field names are added at the beginning of each cell value and are separated by $|$. This transformation makes the link between the field name and the cell value. Instead of only modelling the relation between cells of a specific row, this transformation also models the relation between field names and their cell value.

$$FCC_i = \text{“}[CLS]\ F^1 \mid c_i^1\ [SEP]\ ...\ [SEP]\ F^p \mid c_i^p\ [SEP]\text{''} \quad (2)$$

Text Template Transform (TT). The idea behind the third transformation (Eq. 3) is to use a pre-trained language model, which is trained on large amounts of text data, to represent the sentence in a semantic space (Borisov et al., 2022). In order to obtain a tabular representation in the semantic space, i.e., to obtain the relation between each feature of the tabular data.

$$TT_i = \text{“}[CLS]\ \textit{the}\ F^1\ \textit{is}\ c_i^1,\ ...\ ,\ \textit{the}\ F^p\ \textit{is}\ c_i^p\ [SEP]\text{''} \quad (3)$$

3.2 Pre-Trained Models

The architecture of the pre-trained model chosen is BERT (Devlin et al., 2019). In this study, we experiment with two pre-trained BERT models: BERT (base, uncased)[2] (trained on BookCorpus (Zhu et al., 2015) and English Wikipedia) and FinBERT (Yang et al., 2020)[3], a pre-trained finance-specific language model (trained on financial corpora composed of Corporate Reports 10k & 10-Q, analyst reports and earnings call transcripts).

3.3 Fine-Tuning Strategies

This section explores two frameworks: self-supervised and supervised, with different fine-tuning strategies.

3.3.1 Self-Supervised Framework

The key idea behind this framework is to model the non-forged data distribution and to detect the forged data when the data deviates from the modelled distribution according to a criterion. First, we divide the dataset into forged cases and non-forged cases. Then, we fine-tune a BERT model only on non-forged rows with two training tasks, in order to model the non-forged row distribution with two self-supervised tasks. The first one is named the *Whole Cell Masking* (**WCM**), and the second one is the *Volume of Hypersphere Minimization* (**VHM**).

Task#1: Whole Cell Masking. Similar to (Herzig et al., 2020), we use the whole cell masking that practically masks the entire cell instead of only a token (word) (Devlin et al., 2019). Let $i \leq n$ consider r_i a selected non-forged row in our dataset. Let $j \leq p$ be a selected column in our dataset. Consider the

[2] https://huggingface.co/bert-base-uncased

[3] https://huggingface.co/yiyanghkust/finbert-pretrain

cell $c_i^j \in r_i$. c_i^j can be composed of one word, subwords or words, depending on the tokenizer used. The whole cell masking strategy is to replace all tokens with $[MASK]$ tokens. Let $t \in c_i^j$, and consider h_i^t the embedding vector of the masked token t given by the output of BERT. Eq. 4 gives the probability distribution over the entire tokenizer vocabulary.

$$\hat{y}_i^t = log(Softmax(Wh_i^t + b)) \tag{4}$$

where W and b is the classifier parameters. Then the loss function is the negative log-likelihood (Eq. 5).

$$\mathcal{L}_{wcm} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{M}\frac{1}{Card(c_i^{\sigma_i(j)})}\sum_{t\in c_i^{\sigma_i(j)}} y_i^t\,\hat{y}_i^t \tag{5}$$

where M is the number of masked cells. σ is an element of the permutation group of p elements. $Card$ is the number of elements of a set. Specifically, it is the number of tokens in a cell. y_i^t is one hot encoding vector. The value 1 is at the coordinate of the original token number that has been masked.

Task#2: Volume of Hypersphere Minimization. Similarly to (Guo et al., 2021), we use the training task named volume of hypersphere minimization. This task uses the contextual embedding of the tokens $[CLS]$ that represents the entire row, noted $h_i^{[CLS]}$, given by the output of BERT. The formula of volume of hypersphere minimization is given by Eq. 6.

$$\mathcal{L}_{VHM} = \frac{1}{n}\sum_{i=1}^{n}||h_i^{[CLS]} - c||_2^2 \text{ where } c = \frac{1}{n}\sum_{i=1}^{n} h_i^{[CLS]} \tag{6}$$

First, the hypersphere's centre, c, is computed by averaging all contextual vectors of the non-forged rows. Then, the task is to minimize the volume of the hypersphere by averaging the square Euclidean distance between the previous hypersphere's centre and the contextual vector of the non-forged rows.

Loss. We use as the loss function the linear combination between the WCM loss and the VHM loss (Eq. 7).

$$\mathcal{L}_{final} = \mathcal{L}_{wcm} + \alpha\mathcal{L}_{VHM} \tag{7}$$

The motivation for choosing this loss is that each task has a crucial role to play in modelling the distribution of non-forged data. On the one hand, the first task models the relationship and the coherence between cells of a specific row and, sometimes, between cells and field names depending on the table serialization presented in Section 3.1. On the other hand, the second task gathers non-forged rows close to each other in the contextual space.

Forgery Criterion. After the fine-tuning step, the model is trained on non-forged rows. Here, we evaluate how the forged rows deviate from the non-forged rows distribution thanks to a loss-based criterion. The strategy is to use the WCM method over each tabular cell and use the trained model to get the output of the masked cell. Hence, we obtain p predicted cell distributions (PCD). The j^{th} PCD of the i^{th} row is given by Eq. 8.

$$PCD_i^j = \{(\hat{y}_i^t,\ y_i^t) \mid t \in c_i^j\} \tag{8}$$

where $\hat{y}_i^t$ is the predicted distribution, and y_i^t is the label. Finally, the forgery detection criterion is based on the loss of whole cell masking task value without mean reduction. We sum all errors the trained model makes on a specific row (Eq. 9).

$$\mathcal{L}_i = -\sum_{j=1}^{p}\sum_{(\hat{y}_i,\ y_i)\in PCD_i^j} y_i\,\hat{y}_i \tag{9}$$

After obtaining all loss values from each row, we use an optimized threshold, noted $thresh_{opt}$, in order to maximize the micro-f1 score on the forged class.

$$Criterion(r_i) = \begin{cases} \text{Forged} & \text{if } \mathcal{L}_i > thresh_{opt} \\ \text{Non Forged} & \text{else.} \end{cases} \tag{10}$$

3.3.2 Supervised Framework

The key idea behind this framework is to fine-tune a BERT model for text classification to classify car insurance claims as forged or non-forged. More precisely, let $i \leq n$, r_i a selected row in our dataset and y_i its label ($y_i = 1$ is the forged class and $y_i = 0$ is the non-forged class). Then, we use table serialization transformation on r_i and obtain a list of tokens, noted t_i. Regardless of the table serialization transformation, t_i begins by $[CLS]$ token. Hence, we use $h_i^{[CLS]}$ the contextual embedding, the output of the $[CLS]$ token given by the BERT model, to do the classification task. The classifier part is a linear layer followed by a sigmoid activation function. The r_i output of our framework is given by Eq. 11.

$$\hat{y}_i = Sigmoid(Wh_i^{[CLS]} + b) \tag{11}$$

Loss. The model is fine-tuned using the task of minimization of the binary cross entropy (BCE) given by Eq. 12.

$$BCE = -\frac{1}{n}\sum_{i=1}^{n} y_i log(\hat{y}_i) + (1 - y_i)log(1 - \hat{y}_i) \tag{12}$$

Table 1: Cell reconstruction results (weighted average). The highest values are in **bold**.

	Method	R@1	P@1	F1@1	R@3	P@3	F1@3	R@5	P@5	F1@5
w/o VHM	BERT + CC	64.9 ± 0.9	67.4 ± 0.3	64.3 ± 0.3	90.8 ± 0.8	89.1 ± 0.3	88.0 ± 0.7	94.9 ± 0.6	94.7 ± 0.1	94.1 ± 0.3
	BERT + FCC	63.2 ± 1.1	66.6 ± 1.0	62.9 ± 1.5	89.6 ± 0.2	88.9 ± 0.3	88.1 ± 0.5	95.5 ± 0.6	94.6 ± 0.2	94.4 ± 0.4
	BERT + TT	65.8 ± 1.2	68.1 ± 0.7	64.5 ± 0.8	90.6 ± 0.9	89.3 ± 0.3	88.4 ± 0.5	95.7 ± 0.6	94.9 ± 0.1	94.5 ± 0.4
	FinBERT + CC	**68.4 ± 0.9**	**70.6 ± 0.2**	**67.5 ± 0.5**	**92.6 ± 1.0**	**90.3 ± 0.1**	**89.6 ± 0.3**	96.2 ± 0.5	95.0 ± 0.1	94.8 ± 0.2
	FinBERT + FCC	67.1 ± 1.0	68.9 ± 0.7	66.1 ± 1.0	92.3 ± 0.5	89.8 ± 0.4	89.5 ± 0.7	**96.5 ± 0.4**	95.0 ± 0.1	**95.0 ± 0.3**
	FinBERT + TT	66.1 ± 0.9	69.0 ± 0.3	65.6 ± 0.4	91.8 ± 0.3	90.1 ± 0.2	**89.6 ± 0.3**	96.0 ± 0.6	**95.1 ± 0.1**	94.7 ± 0.2
w/ VHM	BERT + CC	64.6 ± 1.3	67.6 ± 0.5	64.3 ± 0.7	90.1 ± 0.7	88.9 ± 0.4	88.1 ± 0.6	95.5 ± 0.4	94.7 ± 0.1	94.4 ± 0.3
	BERT + FCC	64.1 ± 0.8	67.6 ± 0.7	63.9 ± 1.0	89.5 ± 0.7	89.0 ± 0.1	88.0 ± 0.4	94.8 ± 0.4	94.5 ± 0.1	94.0 ± 0.2
	BERT + TT	65.6 ± 1.3	68.6 ± 0.3	65.4 ± 0.6	90.4 ± 1.0	89.5 ± 0.1	88.7 ± 0.4	95.5 ± 1.1	94.9 ± 0.1	94.6 ± 0.6
	FinBERT + CC	68.1 ± 0.8	70.7 ± 0.3	67.6 ± 0.4	92.3 ± 1.1	90.1 ± 0.1	89.3 ± 0.4	96.2 ± 0.7	95.0 ± 0.1	94.6 ± 0.4
	FinBERT + FCC	67.0 ± 0.7	69.2 ± 0.5	66.2 ± 0.6	91.4 ± 0.7	89.9 ± 0.3	89.2 ± 0.6	95.7 ± 0.8	95.0 ± 0.1	94.6 ± 0.4
	FinBERT + TT	67.2 ± 0.8	69.4 ± 0.4	66.3 ± 0.6	91.2 ± 1.2	90.1 ± 0.3	89.5 ± 0.6	96.4 ± 0.5	95.0 ± 0.1	94.8 ± 0.3

Forgery Criterion. After the fine-tuning step, the forgery detection criterion is based on the probability of the BERT output (Eq. 13).

$$Criterion(r_i) = \begin{cases} \text{Forged} & \text{if } \hat{y}_i > thresh_{opt} \\ \text{Non-forged} & \text{else.} \end{cases} \tag{13}$$

Usually, a threshold of 0.5 is used as a decision criterion. However, in our case, the dataset used is highly imbalanced. Thus, we decided to use an optimized threshold, noted $thresh_{opt}$, on validation data in order to maximize the micro F1 of the forged class.

4 EXPERIMENTAL SETUP

Dataset. We base our study on the real-world car insurance claims dataset provided by *Angoss Knowledge Software* (Phua et al., 2004), which contains 15,420 observations, with 14,497 non-fraudulent and 923 fraudulent rows in tabular format[4]. Each record, represented by a row, contains a set of attributes of an insurance company's customer related to their sociodemographic profile and the insured vehicle. There are seven numerical attributes and twenty-five categorical attributes, and each attribute is pre-sociodemographic.

Dataset Preprocessing. We standardized values for consistent word embeddings. Abbreviations in fields like *Month* were expanded (e.g., "January" for "Jan"). For categories with numerical intervals, such as *Number of supplements*, we replaced "none" with "0". Misspellings, such as "Porche" in the *Make* category, were corrected to "Porsche". The *PolicyType* category, a combination of *BasePolicy* and *VehicleCategory*, had 4,849 mismatches. We split *PolicyType* to update the content of the other two categories and subsequently removed it (Abakarim et al., 2023).

Training Strategy, Metrics, and Hyperparameters. We split the dataset 80:20 and utilized 5-fold cross-validation. Evaluation metrics include precision (P), recall (R), specificity, and P@k, R@k, and F1@k for tabular experiments. Models had a learning rate of 10^{-5}, a batch size of 32, and ran for 10 epochs. In the self-supervised framework, we masked five cells randomly and used an $\alpha = 0.1$ for the final loss calculation (see Eq. 7).

5 RESULTS AND ANALYSIS

In this section, we performed three experiments to evaluate the proposed frameworks. The first experiment allows the evaluation of the ability of a self-supervised framework to model the tabular data (*Cell Reconstruction*); the second, the capacity of language models to detect forged data from non-forged data (*Forgery Detection*); and the third, the loss functions that mitigate the effect of class imbalance during the fine-tuning step (*Loss Ablation Study*).

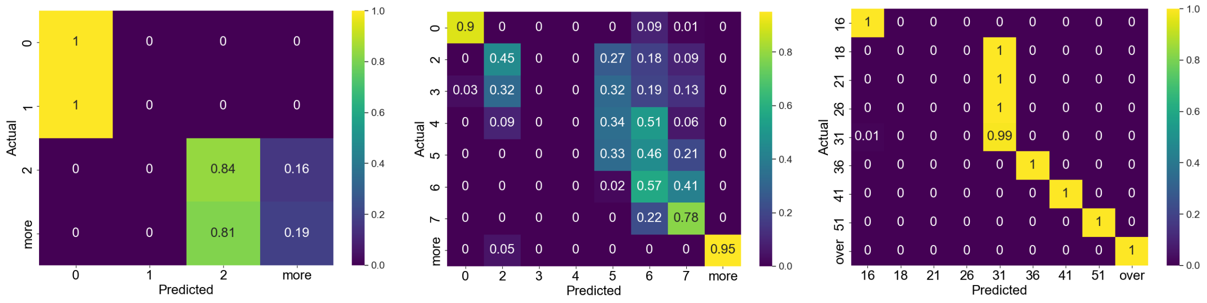
5.1 Cell Reconstruction

In this section, we evaluate the ability of BERT models to reconstruct a cell of the table when it is masked in input based on the semantics of the surrounding cells. A fortiori, this ability is used in the forgery detection criterion. All results are reported in Table 1.

Main Findings. First, we notice that the main drawback is the bias introduced by the training task. When a cell is masked, it can be composed of one or several mask tokens. This number helps the model select specific inter-classes in the field.

More specifically, in the field *PastNumberOfClaims*, with intra-classes like "0", "1", "2 to 4", and "more than 4", we observed token swaps between "0" and 1"", and between "2" and "more" (Figure 2a). Such token permutations are due to differences in

[4] The dataset is available on Kaggle (data science competition platform) at https://www.kaggle.com/datasets/khusheekapoor/vehicle-insurance-fraud-detection.



(a) Past number of a claim. (b) Age of vehicle. (c) Age of policyholder.

Figure 2: First-word confusion matrices of three different fields.

masking lengths for these categories. Similar issues appear in fields like *AgeOfVehicle* (Figure 2b) and others listed. Despite this, the training task helps the model recognize word order in multi-token cells, such as *AgeOfPolicyholder* (e.g. the numbers "18", "21", "26" are unrecognized as "31"). It ensures the correct sequencing of numbers within categories and consistently predicts using field-specific vocabularies.

Second, even if FinBERT is pre-trained on financial documents, it can cause several issues when reconstructing the numerical values (e.g., the fields *RepNumber*, *Age*, *DriverRating* and *Year* have low $F1@1$ score between 0% and 23.1%). However, these difficulties may also be due to the inability of the model to model specific fields from the surrounding fields.

Third, intra-field class imbalance significantly impacts our experimental results. Fields such as *DayPolicyClaim*, *DayPolicyAccident*, and others listed have an $F1@1$ score above 84.6%. However, the majority class in these fields constitutes over 90% of the data, skewing results. For instance, in *DayPolicyClaim*, "more than 30" accounts for 99% of entries, leading to high scores due to the disproportionate representation.

W/o VHM versus W/ VHM. We observe that the minimization of the volume of the hypersphere does not improve or degrade the model performances of model reconstruction when we compare pairwise, i.e. the same model with the same table serialization, but the models trained with and without the VHM task. The performance metrics are very close to each other. This observation allows us to keep this learning task because if it had degraded the quality of the modelling, we would have had problems with the forgery detection step.

BERT versus FinBERT. Independently of table serialization, FinBERT has better results than the BERT (base). If we consider the perfect cell reconstruction (i.e. R@1, P@1 and F1@1), FinBERT increases the overall performance metrics by around 4%. Then, the higher the tolerance threshold, the less FinBERT offers better results than BERT (the improvement is around 1%).

Table Serialization. On the one hand, the TT transformation allows the BERT (base) to perform better results than the other transformations. On the other hand, the CC transformation allows FinBERT to perform better than the other transformations, with an increase of around 3%.

Overall. The best framework composition is based on FinBERT and CC transformation. It must be emphasized that the overall results are very close. It reconstructs the cell with an R@1, P@1, and F1@1 of approximately 68.5%. When the tolerance threshold increases, it allows obtaining metric values around 95%. Thus, the self-supervised framework could be used to model tabular data.

5.2 Forgery Detection

In this section, we evaluate the self-supervised and supervised frameworks' ability to model tabular data to detect forged data while comparing our results with SoTA methods.

BERT versus FinBERT. Globally, whether it is the supervised or the self-supervised framework, the pre-trained FinBERT model provides the best results. Indeed, in the case of supervised learning, the FinBERT model with cell concatenation transformation obtains the best results with high specificity and sensitivity scores.

Fine-Tuning Strategies. When we compare the supervised method and the self-supervised method, the first greatly outperforms the second with a 14% improvement in specificity, a 69% improvement in sensitivity, a 190% improvement in precision and a 154%

Table 2: Results for the supervised and self-supervised frameworks. The highest values per section (SoTA, supervised/self-supervised w/ VHM and w/o VHM) are in **bold**, while the overall highest performance values are underlined.

		Method	Specificity	Sensitivity	Precision	F1
SoTA		(Farquad et al., 2012)	56.22	85.18	-	-
		(Sundarkumar and Ravi, 2015)	58.39	91.89	-	-
		(Nian et al., 2016)	52.00	91.00	-	-
		(Itri et al., 2019b)	-	23.83	**19.66**	**21.52**
		(Majhi et al., 2019)	70.39	**97.47**	-	-
		(Subudhi and Panigrahi, 2020)	**88.45**	83.21	-	-
		(Kate et al., 2023)	57.5	96.0	-	-
Supervised		BERT + CC	86.1 ± 6.9	47.0 ± 13.5	19.2 ± 2.3	26.5 ± 1.0
		BERT + FCC	89.0 ± 7.9	42.3 ± 17.6	22.6 ± 3.7	27.4 ± 3.8
		BERT + TT	86.9 ± 2.5	50.3 ± 8.8	20.3 ± 0.7	28.7 ± 1.4
		FinBERT + CC	**89.7 ± 2.3**	46.0 ± 9.2	**22.9 ± 1.2**	**30.2 ± 2.3**
		FinBERT + FCC	85.8 ± 4.4	**53.3 ± 13.9**	20.1 ± 1.3	28.7 ± 1.7
		FinBERT + TT	87.1 ± 7.0	46.8 ± 18.0	20.5 ± 2.7	27.0 ± 3.0
Self-supervised	w/o VHM	BERT + CC	62.9 ± 14.3	43.1 ± 16.3	7.2 ± 0.2	11.9 ± 1.0
		BERT + FCC	65.2 ± 8.0	42.0 ± 8.8	7.4 ± 0.4	12.5 ± 0.7
		BERT + TT	70.7 ± 12.5	36.9 ± 12.8	7.9 ± 0.5	12.7 ± 0.3
		FinBERT + CC	65.6 ± 20.3	40.9 ± 19.6	7.9 ± 1.1	12.5 ± 0.4
		FinBERT + FCC	**73.9 ± 7.9**	36.8 ± 10.6	**8.5 ± 0.4**	**13.6 ± 1.3**
		FinBERT + TT	64.5 ± 7.1	**45.5 ± 9.0**	7.8 ± 0.4	13.3 ± 0.9
	w/ VHM	BERT + CC	72.0 ± 12.7	34.6 ± 14.0	7.7 ± 0.7	12.3 ± 0.4
		BERT + FCC	59.4 ± 28.3	46.4 ± 25.6	7.5 ± 0.8	12.3 ± 0.8
		BERT + TT	73.2 ± 10.8	31.8 ± 11.0	7.5 ± 0.7	1.8 ± 0.7
		FinBERT + CC	50.5 ± 25.3	**55.0 ± 22.6**	7.1 ± 0.7	12.3 ± 0.6
		FinBERT + FCC	63.9 ± 22.1	43.0 ± 21.6	7.7 ± 0.9	**12.5 ± 0.8**
		FinBERT + TT	**78.6 ± 7.7**	27.2 ± 8.4	**7.9 ± 0.7**	11.9 ± 0.7

improvement in F1. In addition, we can see the importance of the VHM task in the self-supervised framework. Thus, this task improves the specificity score by 6% but reduces the sensitivity by 26%. Even if the decrease in the sensitivity score is important, this task allows for reducing the number of false positives, which is ideal.

Overall. The best results are obtained by the FinBERT model trained in a supervised manner with cell concatenation transformation with a specificity score of 89.7%, sensitivity score of 46.0%, Precision score of 22.9% and F1 score of 30.2%.

Comparison with SoTA. Generally, the methods we explored yielded distinct results compared to others, as seen in Table 2. While other methods prioritize detecting forged data, achieving sensitivity scores between 85.18% and 97.47%, their specificity ranges only from 52.00% to 70.39%, resulting in a high false positive rate. Given a class imbalance of 91:9, they often misclassify non-forged data. Our approach aligns more with (Subudhi and Panigrahi, 2020; Itri et al., 2019a), emphasizing specificity over sensitivity to reduce false positives. Compared to (Itri et al., 2019a), our methods display higher precision and F1, but sensitivity scores lag behind those in (Subudhi and Panigrahi, 2018).

5.3 Loss Ablation Study

The main objective of these experiments is to improve the classification rate of forged data of the supervised framework by using losses designed to change the contribution of each example, depending on its class, to mitigate the effect of class imbalance. Using the same notation as in Section 3, we study three losses to compare their ability to reduce the effect of class imbalance and compare to binary cross entropy (BCE) (Eq. 12) and mean square error (MSE) (Eq. 14).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{14}$$

The first loss studied is the weighted mean square error (WMSE) (Eq. 15). The value of the parameter α_{y_i} is high for the minority class and is low for the majority class to balance the importance of each class during the training phase.

$$WMSE = \frac{1}{n}\sum_{i=1}^{n}\alpha_{y_i}(y_i - \hat{y}_i)^2 \tag{15}$$

Next, we experiment with the dice loss (Li et al., 2019) (Eq. 16) that has as its objective to give more importance during the training process to the minority class (forged) and less to the majority class (non-forged) with the addition of the hyperparameter γ used to smooth the loss and also used by the model to train

Table 3: Ablation study results. The highest values are in **bold**.

Metrics	**MSE**	**WMSE**	**Dice Loss**			**Self-adjust Dice Loss**					
Parameter(s)	N/A	N/A	$\gamma = 10^{-1}$	$\gamma = 10^{-2}$	$\gamma = 10^{-3}$	$\alpha = 10^{-1}$			$\alpha = 10^{-2}$		
						$\gamma = 10^{-1}$	$\gamma = 10^{-2}$	$\gamma = 10^{-3}$	$\gamma = 10^{-1}$	$\gamma = 10^{-2}$	$\gamma = 10^{-3}$
Specificity	**91.6 ± 0.2**	88.8 ± 1.5	91.5 ± 2.4	89.5 ± 1.7	59.8 ± 17.7	36.3 ± 31.8	30.5 ± 29.0	18.1 ± 35.5	51.5 ± 28.5	53.9 ± 28.3	34.5 ± 42.2
Sensitivity	37.2 ± 4.5	41.6 ± 5.2	32.9 ± 7.5	38.8 ± 3.6	68.7 ± 23.7	65.2 ± 29.8	78.5 ± 19.3	**82.8 ± 33.2**	57.0 ± 23.7	52.2 ± 25.9	67.8 ± 39.2
Precision	**21.9 ± 2.8**	19.1 ± 1.2	20.1 ± 2.7	19.2 ± 1.2	10.4 ± 1.7	6.4 ± 0.7	7.3 ± 1.7	6.5 ± 0.8	7.7 ± 1.8	7.1 ± 0.9	6.9 ± 1.2
F1	**27.6 ± 3.4**	26.0 ± 1.2	24.5 ± 2.1	25.6 ± 0.8	17.4 ± 2.3	11.2 ± 0.7	13.2 ± 2.5	11.3 ± 0.4	13.1 ± 2.2	12.1 ± 1.0	11.5 ± 1.2

on the majority class, with squared $\hat{y}_i$ and y_i^2 to accelerate the convergence.

$$DL = \frac{1}{n}\sum_{i=1}^{n} 1 - \frac{2\,\hat{y}_i\,y_i + \gamma}{\hat{y}_i^2 + y_i^2 + \gamma} \qquad (16)$$

Finally, the self-adjust dice loss (Eq. 17) replaces $\hat{y}_i$ by $(1-\hat{y}_i)^{\alpha}\hat{y}_i$ to push down the weight of easy examples.

$$SADL = \frac{1}{n}\sum_{i=1}^{n} 1 - \frac{2(1-\hat{y}_i)^{\alpha}\hat{y}_i\,y_i + \gamma}{(1-\hat{y}_i)^{\alpha}\hat{y}_i + y_i + \gamma} \qquad (17)$$

5.3.1 Results and Analysis

We examine the impact of losses to prevent overfitting on the non-forged class and contrast these findings with the top results from Section 5.2.

MSE vs. WMSE. We set $\alpha_0 = \frac{15420}{15420-923} \approx 1$ and $\alpha_1 = \frac{15420}{923} \approx 16$. Weighting the MSE loss implies that the model will learn less on the non-forged examples in comparison to the usual MSE, and this effect can be observed in the results presented in Table 3. On the one hand, the specificity score of the model trained with WMSE decreases in comparison with that of the model trained with MSE. The specificity score drops from 91.6% to 88.8% On the other hand, the sensitivity score of the model trained with WMSE increases in comparison with that of the model trained with MSE. The specificity score increases from 37.2% to 41.6%.

Dice Loss. We notice that reducing the hyperparameter γ implies that the model learns less on the non-forged examples, i.e. the specificity decreases when gamma decreases. This score drops from 91.5% to 59.8% (Table 3). However, the model learns more about forged examples, and this effect can be observed in Table 3 where the sensitivity score increases from 32.9% to 68% while gamma decreases.

Self-Adjusting Dice Loss. We experiment with different γ and α values, and we notice that the gamma affects the model training the same as that observed with the dice loss. In addition, we observe that the alpha parameter allows during the model training to continue to learn on the non-forged class and avoid the over-fitting on the forged class.

Overall. The results in terms of F1 are lower than the best result given by the loss CE in Section 5.2 (with an F1 of 30.2 ± 2.3 against an F1 between 11.3 ± 0.4 and 26.0 ± 1.2). In addition, the more the models learn about the forged data, the more the variability level increases (e.g. reducing gamma in the dice loss implies that the standard deviations of different metrics increase in Table 3). This highlights the model's difficulty in learning to classify forged from non-forged data.

6 CONCLUSIONS

Our experiments indicate that FinBERT with cell concatenation excels in modelling tabular data through our self-supervised framework. For detecting forged claims, the standout is a supervised FinBERT achieving 89.7% specificity and 46.0% sensitivity, comparable to state-of-the-art results. In the self-supervised setup, the distinction between forged and authentic data is not pronounced, underscoring the challenge of distinguishing them even when guiding the model towards forged data. However, these experiments allowed us to highlight the difficulty of separating the forged data from the non-forged ones by using various losses and forcing the model learning to focus on the forged data.

ACKNOWLEDGEMENTS

This work was supported by the French government in the framework of the France Relance program and by the Itesoft company under grant number AD 22-252.

REFERENCES

Abakarim, Y., Lahby, M., and Attioui, A. (2023). A bagged ensemble convolutional neural networks approach to recognize insurance claim frauds. volume 6.

Abdallah, A., Maarof, M. A., and Zainal, A. (2016). Fraud detection system: A survey. *Journal of Network and Computer Applications*, 68:90–113.

Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., Elhassan, T., Elshafie, H., and

Saif, A. (2022). Financial fraud detection based on machine learning a systematic literature review. *Applied Sciences*, 12(19):9637.

Aslam, F., Hunjra, A. I., Ftiti, Z., Louhichi, W., and Shams, T. (2022). Insurance fraud detection: Evidence from artificial intelligence and machine learning. *Research in International Business and Finance*, 62:101744.

Badaro, G. and Papotti, P. (2022). Transformers for tabular data representation: a tutorial on models and applications. *Proceedings of the VLDB Endowment*, 15(12):3746–3749.

Borisov, V., Broelemann, K., Kasneci, E., and Kasneci, G. (2023). Deeptlf: robust deep neural networks for heterogeneous tabular data. *International Journal of Data Science and Analytics*, 16(1):85–100.

Borisov, V., Seßler, K., Leemann, T., Pawelczyk, M., and Kasneci, G. (2022). Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280*.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., and Bontempi, G. (2015). Credit card fraud detection and concept-drift adaptation with delayed supervised information. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Farquad, M. A. H., Ravi, V., and Raju, S. B. (2012). Analytical crm in banking and finance using svm: a modified active learning-based rule extraction approach. *International Journal of Electronic Customer Relationship Management*.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37.

Gomes, C., Jin, Z., and Yang, H. (2021). Insurance fraud detection with unsupervised deep learning. *Journal of Risk and Insurance*, 88(3):591–624.

Guo, H., Yuan, S., and Wu, X. (2021). Logbert: Log anomaly detection via bert. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Gupta, R., Mudigonda, S., and Baruah, P. K. (2021). Tgans with machine learning models in automobile insurance fraud detection and comparative study with other data imbalance techniques. *International Journal of Recent Technology and Engineering*, 9:236–244.

Hassan, A. K. I. and Abraham, A. (2016). Modeling insurance fraud detection using imbalanced data classification. In *Advances in nature and biologically inspired computing*, pages 117–127. Springer.

Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., and Eisenschlos, J. (2020). TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Itri, B., Mohamed, Y., Mohammed, Q., and Omar, B. (2019a). Performance comparative study of machine learning algorithms for automobile insurance fraud detection. In *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, pages 1–4. IEEE.

Itri, B., Mohamed, Y., Mohammed, Q., and Omar, B. (2019b). Performance comparative study of machine learning algorithms for automobile insurance fraud detection. In *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, page 1–4.

Jiao, Q. and Zhang, S. (2021). A brief survey of word embedding and its recent development. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, volume 5, page 1697–1701.

Kate, P., Ravi, V., and Gangwar, A. (2023). Fingan: Chaotic generative adversarial network for analytical customer relationship management in banking and insurance. *Neural Computing and Applications*, 35(8):6015–6028.

Kirlidog, M. and Asuk, C. (2012). A fraud detection approach with data mining in health insurance. *Procedia-Social and Behavioral Sciences*, 62:989–994.

Krawczyk, B. and Woźniak, M. (2015). One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing*, 19(12):3387–3400.

Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., and Li, J. (2019). Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*.

Majhi, S. K., Bhatachharya, S., Pradhan, R., and Biswal, S. (2019). Fuzzy clustering using salp swarm algorithm for automobile insurance fraud detection. *Journal of Intelligent & Fuzzy Systems*, 36(3):2333–2344.

Nian, K., Zhang, H., Tayal, A., Coleman, T., and Li, Y. (2016). Auto insurance fraud detection using unsupervised spectral ranking for anomaly. *The Journal of Finance and Data Science*, 2(1):58–75.

Peng, Y., Kou, G., Sabatka, A., Chen, Z., Khazanchi, D., and Shi, Y. (2006). Application of clustering methods to health insurance fraud detection. In *2006 International Conference on Service Systems and Service Management*, volume 1, pages 116–120. IEEE.

Phua, C., Alahakoon, D., and Lee, V. (2004). Minority report in fraud detection: Classification of skewed data. *SIGKDD Explor. Newsl.*, 6(1):50–59.

Ryman-Tubb, N. F., Krause, P., and Garn, W. (2018). How artificial intelligence and machine learning research impacts payment card fraud detection: A survey and industry benchmark. *Engineering Applications of Artificial Intelligence*, 76:130–157.

Sithic, H. L. and Balasubramanian, T. (2013). Survey of insurance fraud detection using data mining techniques. *arXiv preprint arXiv:1309.0806*.

Soufiane, E., EL Baghdadi, S.-E., Berrahou, A., Mesbah, A., and Berbia, H. (2022). Automobile insurance claims auditing: A comprehensive survey on handling awry datasets. In *WITS 2020: Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems*, pages 135–144. Springer.

Subudhi, S. and Panigrahi, S. (2018). Detection of automobile insurance fraud using feature selection and data mining techniques. *International Journal of Rough Sets and Data Analysis (IJRSDA)*, 5(3):1–20.

Subudhi, S. and Panigrahi, S. (2020). Use of optimized fuzzy c-means clustering and supervised classifiers for automobile insurance fraud detection. *Journal of King Saud University - Computer and Information Sciences*, 32(5):568–575.

Sundarkumar, G. G. and Ravi, V. (2015). A novel hybrid undersampling method for mining unbalanced datasets in banking and insurance. *Engineering Applications of Artificial Intelligence*, 37:368–377.

Tennyson, S. and Salsas-Forn, P. (2002). Claims auditing in automobile insurance: fraud detection and deterrence objectives. *Journal of Risk and Insurance*, 69(3):289–308.

Yang, Y., Uy, M. C. S., and Huang, A. (2020). Finbert: A pretrained language model for financial communications. *arXiv preprint arXiv:2006.08097*.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.