An Explainable Attention Network for Fraud Detection in Claims Management

Helmut Farbmacher
Max Planck Society, Germany

Leander Löw
University of Hamburg, Germany

Martin Spindler
University of Hamburg, Germany

Insurance companies must manage millions of claims per year. While most of these claims are not fraudulent, those that are nevertheless cost insurance companies and those they insure vast amounts of money. The ultimate goal is a predictive model to single out fraudulent claims and pay out non-fraudulent ones automatically. Health care claims, however, have a peculiar data structure, which is hierarchical and of variable length. We exploit similarities between the structure of a health care claim and the structure of a text, and extend deep learning models from text classification to claims management. Using a dataset of two million claims from a private health insurer in Germany, we show that our proposed models outperform bag-of-words based models, hand-designed features, and models based on convolutional neural networks. We also investigate the extent to which our models deliver meaningful explanations for their predictions. Meaningful explanations advance fraud detection models from methods that are merely predictive to those that are prescriptive.

Key words: Claims management, health insurance, machine learning, fraud detection, prescriptive analytics

1. Introduction

Health insurers receive millions of claims per year. Given that information asymmetries between the principal (insurer) and the agents (health care providers and the insured) can lead to moral hazard, insurance companies face the choice of either paying out insurance claims immediately without any adjustments or reviewing suspicious claims. The most common method for undertaking the latter involves manually auditing claims data, which is a time-consuming and expensive process (see, for instance, Townsend 1979, Bond and Crocker 1997). Machine-learning models, however, can greatly cut auditing costs by automatically screening incoming claims and flagging up those that are deemed to be suspicious – i.e., potentially incorrect – for subsequent manual auditing.

Health care claims have an unusual input data structure. At each doctor consultation or hospital visit, several tasks are performed and may be billed separately. Medical coders identify all services, prescriptions, and supplies received during the insured's appointment and create an insurance claim. Each claim consists of multiple items, including the dates of treatment, a short description of the task or tasks performed, and the associated costs. The number of items varies from patient to patient, resulting in claims of variable length. A fully-fledged machine-learning model must take this hierarchical data structure into account. In our models, we do so by extending recent developments in text classification.

Because ordinary machine-learning methods, such as gradient boosting or random forests, require a vector of fixed size as input, the standard approach so far has been to find a fixed-size vector by manually engineering features based on domain knowledge. Doing so, however, requires costly domain experts and does not scale well to other problems, even if they are similar. Moreover, good features can be difficult to build with manual feature engineering. In search of a better solution and, in particular, a way to automate this process, we exploit the similarities between the data structure of a claim and the data structure of a text. Indeed, both consist of a sequence of vectors for each observation, which, in the case of a text, is a sequence of words and, in the case of claims, is a sequence of tasks.

In recent years, machine-learning methods based on artificial neural networks have begun to show human-level performance in text processing (see, for example, Kim 2014, Lai et al. 2015, Vaswani et al. 2017, Devlin et al. 2018). Because methods like these can directly handle unusual data structures and perform feature engineering as part of the learning process, they may represent an automated alternative to manual feature engineering. We pursue this approach for an important step in the claims management process: auditing and managing the risk of fraud and error. Additionally, we investigate the extent to which such models deliver meaningful explanations for their predictions. These explanations can be used to shed light on suspicious claim items, greatly simplifying the subsequent auditing process. Meaningful explanations, therefore, advance machine-learning models from ones that are merely predictive to ones that are prescriptive.

In our empirical application, we train our machine-learning models on real data containing past claims and their associated labels, which classify each claim as either correct or suspicious. The latter may refer to any kind of error, for example due to unintentional upcoding or fraud. For a comprehensive overview of various fraud behaviors see Li et al. (2008). For the sake of simplicity, we refer in the following to the detection of suspicious errors as (insurance) fraud detection. For each claim, we want to predict the probability of fraud and pay out all non-fraudulent claims automatically without any further manual auditing. Fraudulent or suspicious claims, however, should be flagged up so that they can be audited. Our supervised machine-learning models are tailored to this task. Like all supervised models, they should be updated regularly to capture new types of fraud and changes in regulations.

There is a vast literature on the economics of auditing and optimal auditing (see, for instance, Townsend 1979, Mookherjee and Png 1989, Picard 1996, Schiller 2006, Dionne et al. 2009, and, for an overview, Picard 2013). Schiller (2006) provided evidence that auditing becomes more effective when insurers condition their audits on the information provided by fraud detection systems. Dionne et al. (2009) showed that an optimal auditing strategy takes the form of a red-flags approach, which entails referring a claim to the auditing unit once certain fraud signals have been observed. We, in turn, demonstrate that our machine-learning models are particularly useful tools in the claims management process because they not only add another red flag but give further insights into the fraud mechanisms themselves. We illustrate this using a Monte Carlo simulation.

Detecting fraud or systemic abuse is a major challenge for health insurers (see, for instance, Becker et al. 2005, Heese 2018, Bastani et al. 2018). There is also a vast literature on data-driven methods for fraud detection. For instance, Liou et al. (2008) compared the accuracy of logistic regression, neural networks and classification trees in identifying fraud and claim abuse in diabetic outpatient services. Viaene et al. (2002) compared the performance of several classification techniques in detecting car insurance fraud. Van Vlasselaer et al. (2017) proposed a model that employs information about firm networks to detect social security fraud. Additionally, classification methods have been used in other settings, for example to detect management fraud (Cecchini et al. 2010) or identify high-risk disability claims (Urbanovich et al. 2003). An overview of data-driven methods to fraud assessment is given by Bolton and Hand (2002) and Ekin et al. (2018), the latter of whom focus on the assessment of medical fraud.

In the next section, we discuss an important trade-off that a cost-minimizing insurer faces in the claims management process. Our economic evaluation criterion, which we use to assess the performance of our machine-learning models, reflects this trade-off. In section 3 we shed further light on the similarities between the classification of text and the classification of claims. Section 4 presents our machine-learning models for claims classification. Section 5 discusses the results of the models when applied to claims data from a private health insurer in Germany. In section 6 we use a Monte Carlo simulation to illustrate how we can derive meaningful explanations for the predictions of our machine-learning models. Section 7 concludes.

2. The Economics of Auditing

A cost-minimizing insurance company faces a trade-off between the expected benefits of auditing (above all, the expected value of the adjustments to upcoded or fraudulent claims) and the auditing costs. This trade-off is an important component in models of optimal auditing (see, e.g., Mookherjee and Png 1989). Our economic evaluation criterion, which we use to evaluate our machine-learning models, reflects this trade-off.

The main goal of our machine-learning models is to identify suspicious claims automatically. A crucial aspect of the models is their ability to predict the probability of fraud (P) for any of the claims in our data. In doing so, two types of error may occur: the model may classify a non-fraudulent claim as suspicious, which would be a false positive, or it might fail to identify a fraudulent claim, which would be a false negative. Our evaluation criterion takes both errors into account. Let N denote the number of fraudulent claims, M the number of false positives, c the fixed costs of manual auditing, and a_i the adjustment associated with a particular claim. The last of these represents the cash benefit for that claim if correctly identified as suspicious. We assume that manual auditing of the suspicious claims detected by our models correctly identifies truly fraudulent claims with a probability equal to one. In this manner, the auditing process conditional on any machine-model model achieves a cost reduction of π as follows:

$$\pi = \sum_{i=1}^{N} I(P_i > \tau) \cdot (a_i - c) - Mc\,. \tag{1}$$

The first term captures the net benefit of the true positives, and the second term the costs of the false positives. $I()$ is the indicator function, which equals 1 if the statement in parentheses is true. τ is a threshold chosen by the insurer, which is usually set at 0.5 but can, in principle, be any other value in between 0 and 1 depending on free capacities in the auditing unit, risk preferences of the principal, or (fraud) signals from agents. Such a signal could, for instance, be based on the total number of claims a health care provider submits to the insurer (Li et al. 2008) or on hours worked by the provider (Fang and Gong 2017). Generally, a lower value of τ increases the number of suspicious claims at the cost of a (potentially) larger fraction of false positives.

To incorporate the cost reductions missed due to false negatives, we compare π with the maximum reduction in cost that can be achieved in the auditing process. This maximum is defined as

$$\pi^{\max} = \sum_{i=1}^{N} (a_i - c)\,. \tag{2}$$

Note that the missing indicator function implies that there are no false negatives. Moreover, there are also no false positives in $\pi^{\max}$. That is, it reflects the maximum cost reduction that we can achieve if we know the fraudulent claims a priori from an oracle model.

Finally, our evaluation criterion is defined as

$$\gamma = \frac{\pi}{\pi^{\max}}\,, \tag{3}$$

which measures the fraction of cost reduction that we are able to achieve with our machine-learning model. γ is a relative, unitless measure of the efficacy of the model. A cost-minimizing insurer aims to set $\pi = \pi^{\max}$.

For simplification, our evaluation criterion abstracts from several issues. First, we assume that the cost of manual auditing is a fixed amount per suspicious claim. However, one can imagine that these costs do not scale to all sizes of auditing unit. A small unit that processes only pre-selected claims, for example, may be less expensive per claim than an auditing unit that has to verify all of the claims an insurer receives. As a result, our evaluation criterion may underestimate the potential benefits of a fraud detection system. Second, the benefits of auditing claims are not limited to the identification and adjustment of fraudulent claims but rather extend to the deterrent effect that auditing can have on potential defrauders. Tennyson and Salsas-Forn (2002) and Dionne et al. (2009) are two of many authors who discuss the importance of claims auditing as a deterrent. Effective machine-learning models could strengthen this effect. Third, meaningful explanations for suspicious claims may considerably shorten the subsequent (manual) auditing process. In summary, our evaluation criterion does not represent a general measure of the benefits of auditing but rather focuses on the direct net gains of our machine-learning models.

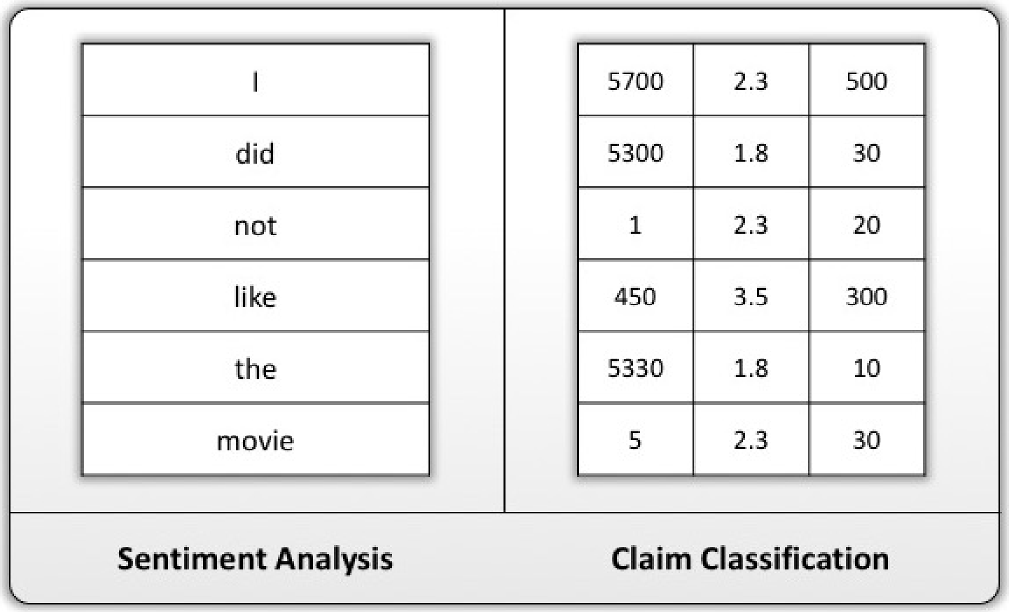
3. Sequence Classification

Auditing can be seen as a sequence classification process: The input is a sequence of claim items and the task is to predict the probability of fraud. In a sequence classification process, as defined in Graves (2012), the task is to predict a fixed-size vector, usually containing probabilities. Generally, items in health care claims consist of several variables, which indicate, for instance, the quantity and type of treatment provided, the potential complications of the treatment, and its price. In our empirical application based on claims data from a private health insurer in Germany, these variables consist of a procedure code, a multiplier and a numerical value, respectively. We can think of the input data structure per claim as a matrix with a variable number of rows. Each triplet of procedure code, multiplier, and price can be considered one row in the data matrix. This way, we interpret claims input data as a sequence of variables similar to text input data, which consist of a sequence of words (see the illustration in Figure 1). In contrast to sequences of words, however, the order of claim items is arbitrary. This arbitrary order is an important feature, which must be incorporated in our machine-learning models.

3.1. Sentiment Analysis

A well-studied sequence classification task is sentiment analysis, which involves predicting the “sentiment” of a statement given a sequence of words. This is usually expressed as the scalar probability of a text expressing a negative or positive sentiment. In the text in Figure 1, for example, the algorithm should be able to identify the negative attitude towards movies. Most machine-learning methods for sentiment analysis rely either on bag-of-words models or neural networks.

Figure 1 Comparison between text input and claim input for a single observation (i.e., a sentence or a claim).



I
did
not
like
the
movie

Sentiment Analysis

5700	2.3	500
5300	1.8	30
1	2.3	20
450	3.5	300
5330	1.8	10
5	2.3	30

Claim Classification

In a bag-of-words model, we create a fixed-size representation of the words by one-hot encoding them and summing them up (see illustration in Figure 2).[1] Afterwards, machine-learning methods (such as a feed-forward network algorithm) can be applied. Bag-of-words models are also applicable to fraud detection, and we will use one as the baseline model in our study.

Methods for sentiment analysis based on neural networks usually rely on four distinct types of layers: The first type is an *embedding layer*, in which the sequence of words has been turned into a sequence of real-valued inputs. One can use either pre-trained word embeddings, as in Mikolov et al. (2013), or learn the embeddings as part of the training process. The second type is a *feature extraction layer*, in which the sequence of inputs has been turned into a sequence of context-dependent representations. Most sequence classification models rely either on recurrent neural networks or convolutional neural networks in this step (Kim 2014). The third type is an *aggregation layer*, which is used to turn the sequence of feature vectors into one fixed-size vector. Most often, max pooling or neural attention are used (**?**). The final type is a *fully connected layer*, which is used to obtain a final prediction from the aggregated feature representation. In this step, sentiment analysis with deep learning uses a feed-forward network to be able to train the whole structure, end to end.

In our application of sentiment analysis to the claims management process, we follow this structure of embedding, feature extraction, aggregation, and fully connected layer. Extending such

[1] The term "one-hot encoding" is generally used in the machine-learning literature to indicate the recoding of a categorical variable into indicator variables for every category.

text classification methods to our claim classification task entails important challenges, however, including the need for a different feature extraction layer, as we describe in the next chapter.

3.2. Extension to Claims Classification

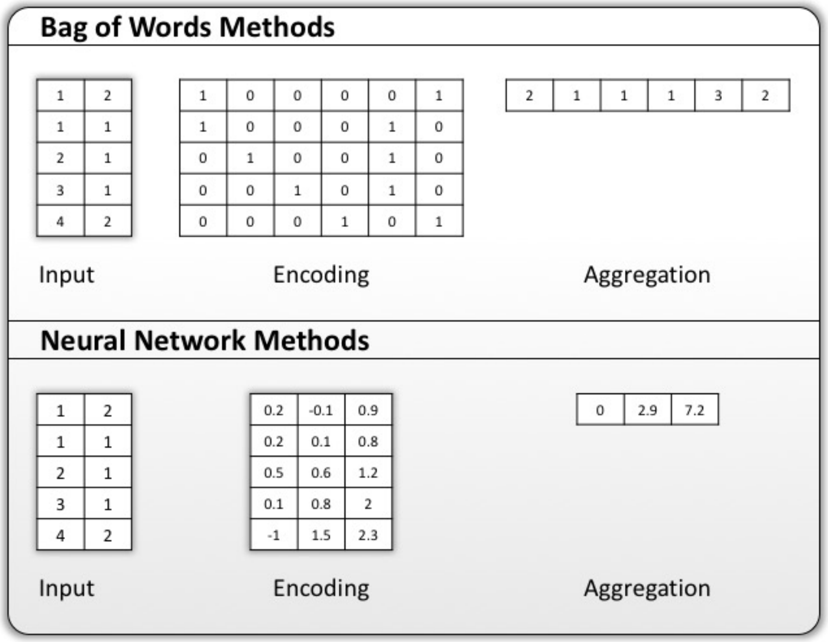
Given the similarity in the data structures, it seems natural to exploit models developed for sentiment analysis for a claims management task. However, whereas words in a statement have a meaningful order, items in a claim do not. This has important consequences for the feature extraction layer. Recall that the purpose of the feature extraction layer is to derive a context-dependent representation – i.e. the "meaning" – of any element of the input sequence. To understand the meaning of a word, the feature extraction mechanism in sentiment analysis must incorporate the order of words. Recurrent neural networks (RNN) exploit the order of words and the position in a text, which makes RNN particularly useful for text processing. Convolutional neural networks (CNN) are another method for text processing. They do not maintain the order of the sequence and are able to consider only local dependency, which means that words close to a target word play a particularly important role in interpreting its meaning.

Due to the arbitrary order of the claim items, we need a model that is invariant to the order of the items but at the same time is able to form context-dependent representations. Both RNN and CNN seem inappropriate for this task. We therefore propose three types of models that are particularly well suited to the claims management process: 1) a model very similar to the classical bag-of-words method, 2) a model based on position-wise feed forward neural networks, and 3) a model based on self-attention neural networks.

As a baseline model, we use a *bag-of-words model*, in which we one-hot encode the categorical variables and sum them up. Similarly, we form the sum of the numerical variable, so that it simply contains the complete costs of a claim. It is important to note, however, that this model loses the association between variables from the same claim item. This is a drawback given that the interactions between variables from the same item may contain important information about upcoding or fraud.

In the context of sequence classification, a *position-wise feed-forward network* can be thought of as a convolutional neural network in which a single claim item represents the context. Unlike the bag-of-words approach, the position-wise feed-forward model retains the relationships between variables from the same claim item since the input of the position-wise network comprises all the variables in a particular claim row. A drawback of this model is that it does not form a context-dependent representation over the entire claim – that is, a certain triplet of procedure code, multiplier, and costs always has the same representation in the feature extraction layer. However, the contribution of a certain item to identifying potential upcoding or fraud may differ depending on the other items

Figure 2 **Comparison between bag-of-words type methods and neural-network-based methods.**



1	2
1	1
2	1
3	1
4	2

1	0	0	0	0	1
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0
0	0	0	1	0	1

2	1	1	1	3	2

1	2
1	1
2	1
3	1
4	2

0.2	-0.1	0.9
0.2	0.1	0.8
0.5	0.6	1.2
0.1	0.8	2
-1	1.5	2.3

0	2.9	7.2

Note. The example has only two categorical variables, one with four and the other with two levels. For the bag-of-words method, we one-hot encode the categorical variables and then sum them up. For neural-network-based methods we "encode" the input variables using an embedding layer followed by a feature extraction layer, yielding a dense representation. Subsequently, we apply the aggregation layer in this dense space.

in a particular claim. We, therefore, conjecture that a context-dependent representation in which the complete list of claim items represents the context of each single item might be advantageous for interpreting an individual claim item.

To illustrate the concept of context representation, we can compare it to texts: In texts, words are almost completely defined by their context. We can see this clearly with words that have multiple meanings. For instance, the word "nails" can mean either finger nails or nails made out of metal. Moreover, negation plays an important role. The word "like" has the exact opposite meaning if it is preceded by the word "not". From these two examples, we can see that context is crucial to the interpretation of language, and thus it is useful to find context-dependent representations.

Self-attention was introduced in Vaswani et al. (2017), where it was applied to the task of translation; Shen et al. (2017), in turn, applied the concept to the task of sequence classification. The main benefit of self-attention networks compared to position-wise feed-forward networks is that they can form a context-dependent representation. The intuition of self-attention is as follows: For each input i, we define an attention distribution over the other inputs. This distribution will give high weight to input $j(\neq i)$, which is particularly relevant for the interpretation of input i. We then use these attention weights in combination with the other inputs to form the derived feature for input i. Hence, the derived features of an input can depend on any other elements of the input sequence, independently of their placement, which means that the order of the items is irrelevant.

3.3. Attention Weights

A deeper analysis of attention weights may contain particularly valuable information for the subsequent claims management process. The weights can be analyzed after estimation to determine which claim items were particularly relevant to the interpretation of a given claim item. We favor a slight modification of the neural attention mechanism in the aggregation layer. As an aggregation method, neural attention can be thought of as a weighted mean of the input data. Usually, these weights are softmax normalized and sum to one. Instead, our modification is to sigmoid normalize them and, hence, they need not sum to one. The motivation behind this is that longer sequences tend to be suspicious more often. Replacing softmax with sigmoid takes account of this by giving a slight nudge towards classifying longer sequences as suspicious cases. We provide more detail about this approach in the following section. Although attention weights are often used to "explain" predictions, recent research has cast doubt on whether they can be interpreted meaningfully (Jain and Wallace 2019). In section 6, we explore the degree to which we can interpret attention weights from the position-wise feed-forward and self-attention models – an important issue in health claims management given that meaningful explanations allow us to advance our models from ones that are merely predictive to ones that are prescriptive.

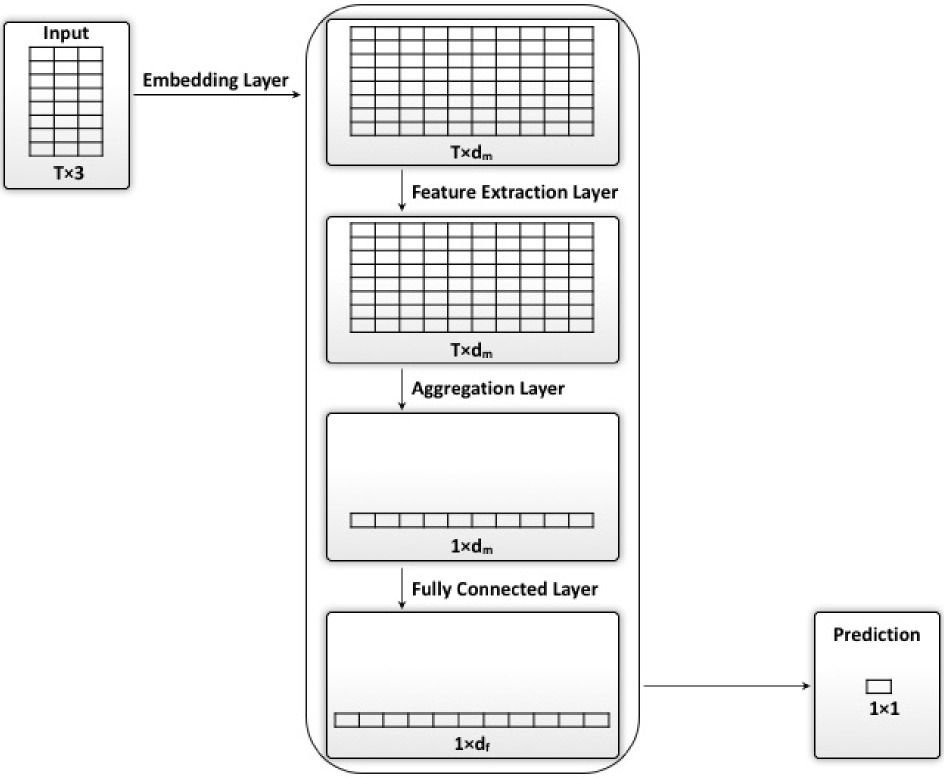
4. Details of the Machine-Learning Model

In this section, we describe our models, which use deep learning to process health care claims and identify suspicious claims automatically. It consists of four distinct types of layers. Figure 3 depicts how a sequence of categorical and numerical variables is turned into a sequence of real vectors by embedding the categorical variables and then applying a feature extraction layer to derive a sequence of features. This sequence is subsequently turned into a fixed-size vector by an aggregation layer. Lastly, we use a fully connected layer to derive a scalar prediction.

4.1. The Embedding Layer

Categorical embeddings are a method for adapting word embeddings to general types of categorical variables, as demonstrated by Guo and Berkhahn (2016). Categorical embeddings associate a dense representation to each claim item. The representation is learned during the training process. The embedding layer is especially useful for our setting due to the high cardinality of the procedure code for medical services. When we one-hot encode this categorical variable, as has been done in simpler models, the high cardinality leads to a huge set of parameters, which in turn can result in overfitting. The intuition behind embeddings is that – instead of using one-hot encoded variables,

Figure 3 Illustration of the distinct layers in neural-network-based models.

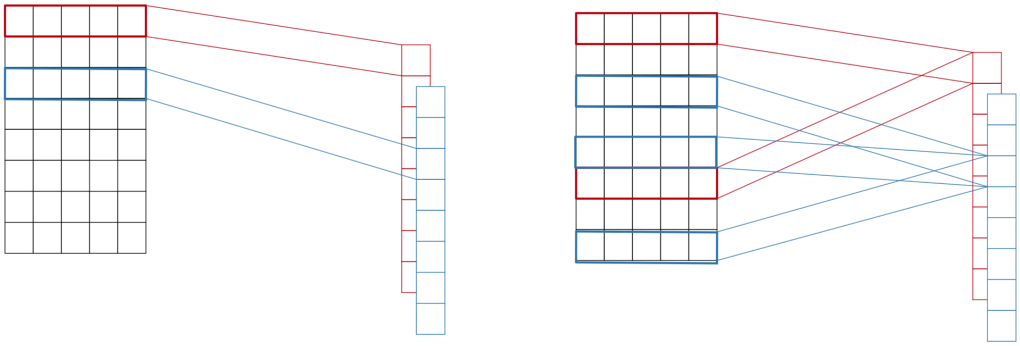


which implies that each input level is orthogonal to every other – we allow for dependence between the levels of categorical variables. Using an embedding layer rather than one-hot encoding reduces the number of parameters substantially and leads to better predictions, albeit at the expense of increased training and inference time.

We use the output of the embedding layer as our input matrix X in the feature extraction layer. The dimension of X is $d_m \times T$, where T denotes the number of claim items (analogous to the number of words in sentiment analysis) and d_m denotes the size of our hidden representation. d_m is a hyperparameter chosen by the analyst.

4.2. The Feature Extraction Layer

This layer is responsible for capturing interactions between parts of the inputs. The feature extraction layer turns the sequence of input variables into a sequence of low-dimensional dense features. We compare two different feature extraction layers: a position-wise feed-forward network and a self-attention network. The main difference between these is that the former produces context-independent representations, whereas the latter can produce representations that incorporate the context. This is illustrated in Figure 4. Whereas a feature of an input element depends, in position-wise networks, only on its input, it depends in self-attention networks both on itself and any other elements of the sequence. Moreover, the degree to which an input element in self-attention networks



(a) Position-wise feed-forward

(b) Self-attention

Figure 4 Comparison between position-wise feed-forward and self-attention feature extractors.

depends on other elements is captured in the attention weights. In this way, the self-attention mechanism can form a context-dependent representation of a claim item.

Position-Wise Feed-Forward:

For each observation, the sequence of inputs represented by matrix X is turned into a sequence of features H by applying a position-wise feed-forward model:

$$H = relu(W_1 X + b_1)$$
$$W_1 \in \mathbb{R}^{d_m \times d_m}; H \in \mathbb{R}^{d_m \times T};\ b_1 \in \mathbb{R}^{d_m}\ .$$

Self-Attention:

For each observation, the sequence of inputs represented by matrix X is turned into a sequence of features H by applying scaled dot-product self-attention, as in Vaswani et al. (2017), whom we follow here. We use a self-attention model with a single head and single layer. First, we derive a set of queries (Q), a set of keys (K), and a set of values (V) from linear transformations of our inputs – via matrix multiplication with parameters and adding a bias, as follows:[2]

$$Q = W_q X + b_q$$
$$K = W_K X + b_k$$
$$V = W_V X + b_v$$
$$W_Q, W_K, W_V \in \mathbb{R}^{d_m \times T},\ b_q, b_k, b_v \in \mathbb{R}^{d_m},\ Q, K, V \in \mathbb{R}^{d_m \times T}.$$

Second, we use the scaled dot-product attention operation to turn the queries, keys, and values into a sequence of features H_j $(j = 1, 2, 3, 4)$:

$$H_1 = softmax(\tfrac{QK^T}{\sqrt{d_m}})V\ ,$$
$$H_1 \in \mathbb{R}^{d_m \times T}\ .$$

Third, we apply a residual and normalization layer as described in Wu et al. (2018) and Ba et al. (2016):

[2] Bias denotes the constant in the neural-network literature.

$$H_2 = layernorm(H_1 + X)\,.$$

Fourth, we apply a position-wise feed-forward layer to these derived features:

$$H_3 = W_2 relu(W_1 H_2 + b_1) + b_2\,,$$

$$W_1 \in \mathbb{R}^{d_m*2\times d_m}; W_2 \in \mathbb{R}^{d_m\times d_m*2}; H \in \mathbb{R}^{d_m\times T}\,.$$

Finally, we apply another normalization and residual layer:

$$H = layernorm(H_2 + H_3)\,.$$

The self-attention algorithm is also summarized in Algorithm 1.

It is crucial that all parameters be independent of the claim length (T). This is a requisite for incorporating variable sequence lengths, and it even enables the model to handle claims of unseen lengths. The features derived from self-attention are context-dependent – that is, we can see which other elements in the input sequence are important for the features of a particular item. In Section 6, we investigate the information content elicited by the features and illustrate how this can be used to increase the efficiency of the manual auditing process.

4.3. The Aggregation Layer

To incorporate variable length sequences efficiently, we must find one fixed-size vector for each possible sequence length. This is the task of the aggregation layer. We begin with the sequence of features ($H \in \mathbb{R}^{d_m\times T}$), which we derive either from position-wise feed-forward or self-attention (as discussed above). The sequence of features has variable length T, and we now want to find a fixed-size representation ($h \in \mathbb{R}^{d_m}$) in the aggregation layer.

Sum, mean, and max pooling are simple methods that either sum, take the average, or identify the maximum value over the corresponding dimension of the feature tensor. Self-attention is a more advanced technique and, put simply, entails taking a weighted average, where the weights are learned in the training data. As mentioned briefly in Section 3.3, we generally observe a positive correlation between suspicious claims and sequence length in claims data. Our aggregation method should therefore be able to scale with the sequence length. For this reason, sum pooling is an obvious choice. We modify sum pooling slightly, and call this form of pooling “sigmoid attention”.

Algorithm 1 Self-Attention

```
  Input: sequence X
  Query, Key, Value: Linear(X)
  $H_1$: Attention(Query, Key, Value)
  $H_2$: Residual+Normalization($H_1 + X$)
  $H_3$: Position-Wise Feed Forward($H_2$)
  $H$: Residual+Normalization($H_2 + H_3$)
  Output: $H$
```

First, we derive the attention weights as a linear layer with sigmoid activation:

$$a = sigmoid(W_a H + b_a)$$

$$H \in \mathbb{R}^{d_m \times T};\; W_a \in \mathbb{R}^{T};\; b_a \in \mathbb{R}^{d_m};\; a \in \mathbb{R}^{T}\,.$$

Next, we use these weights to form a weighted sum of the elements of the feature sequence to derive a fixed-size vector:

$$h = Ha$$

$$h \in \mathbb{R}^{d_m}\,.$$

In contrast to attention with the softmax operator, our attention weights are generated by the sigmoid operator with weights that do not sum up to one. Similar to ordinary attention, this enables us to interpret the attention weights of each claim to identify items that are particularly relevant for classifying a claim as suspicious. We analyze the relative importance of these weights for the subsequent auditing decision in Section 6.

4.4. The Fully Connected Layer

After the aggregation layer, we have one vector of fixed dimension per claim (h) and the corresponding label for this claim. This leaves us with what essentially is a "standard" machine-learning problem. In this step, we could, for instance, use random forests (Breiman 2001), boosted trees (Breiman et al. 1984), or a feed-forward network. We decided to set up a feed-forward network in this last step to train the whole architecture, end to end. In our case we therefore obtain the scalar predictions (P) from the fixed-size aggregation h using a two-layer fully connected neural network, as follows:

$$h_1 = relu(W_1 h + b_1)$$

$$h_2 = relu(W_2 h_1 + b_2)$$

$$P = relu(W_3 h_2 + b_3)$$

$$W_1 \in \mathbb{R}^{d_f \times d_m};\; W_2 \in \mathbb{R}^{d_f \times d_f};\; W_3 \in \mathbb{R}^{1 \times d_f};$$

$$b_1 \in \mathbb{R}^{d_f};\; b_2 \in \mathbb{R}^{d_f};\; b_3 \in \mathbb{R};\; h_1, h_2 \in \mathbb{R}^{T}\,.$$

5. Empirical Evaluation

In our empirical study, we employ two million health care claims from a private health insurer in Germany. Generally, items in health care claims consists of several variables, which indicate the quantity and type of medical treatment provided, potential complications and the price of the treatment. Depending on the type of health insurance and the country in question, prices will vary for the same service and may thus contain additional information about the potential for settling suspicious claims. This is, for instance, the case for private health insurance in the US. Alternatively, there may be a physician fee schedule, which determines a fixed price for any medical service (as, for instance, in Medicare in the US or generally in private and public health insurance

Table 1 **Hyperparameters per model.**

Model	d_m	d_f	Dropout	Weight Decay
CNN	64	512	0	$1e07$
BOW	-	1024	0.1	$1e05$
PFF	32	512	0	$1e06$
SelfA	128	512	0	$1e05$

in Germany). As noted above, private health care claims in Germany contain three variables: a procedure code specifying which medical service was provided, a multiplier, which is a measure of potential complications, and the price of the medical service in question (as a numerical value). Each procedure code is assigned a base price, which, after the multiplier has been applied, yields the price of this medical service.

For each claim, we have a sequence of input vectors ranging from 1 to 100 items. Each input vector consists of three variables: (1) the procedure code, which is a categorical variable with over $4,000$ categories; (2) a factor variable with six categories measuring the patient-specific severity of the task; and (3) a numeric variable (price), which we scale between zero and one after log transformation. We can interpret each input as a two-dimensional matrix, where each row is one claim item consisting of these three variables. Each claim has an associated label indicating whether an adjustment was made to it. For example, if the original charges in a claim were 100€ but ultimately only 80€ were paid, then we define the difference as the adjustment and label the claim as suspicious. We use this actual adjustment to scale the loss function.

Because of the imbalanced response variable, standard evaluation metrics like accuracy and cross-entropy loss are not suitable for model evaluation. The default metrics for imbalanced problems are the area under the receiver operator curve (AUROC) or the area under precision recall curve (AUPR). For our particular application, we additionally use the evaluation criterion discussed in Section 2, which captures the trade-off between the expected benefits of auditing and the auditing costs.[3] All metrics are evaluated on an independent test dataset.

For all models, we use the Adam optimizer (Kingma and Ba 2014) with learning rate $1e04$, weight decay according to Table 1, early stopping, and validation loss as the stopping criterion. As our loss in the training step, we employ a scaled cross-entropy loss function, where the sample weights for loss scaling are the adjustment divided by a normalizing factor. We selected the hyperparameters by random search. The chosen hyperparameters are displayed in Table 1. We use dropout in the bag-of-words model as this controls for overfitting by artificially corrupting the training data (Wager et al. 2013). We train on batches of size 128 with claims of the same size and implement the models in Keras.

[3] We use an internal estimate for the fixed costs of manual auditing (c). In robustness checks we observed that the relative performance of our machine-learning models does not depend on this estimate.

Table 2 Model Comparison.

MODEL	AUROC	AUPR	γ
CNN	0.902	0.195	0.661
MANUAL + BOOSTED TREES	0.912	0.231	0.680
BOW	0.911	0.227	0.673
PFF	0.923	0.251	0.713
SELFA	0.926	0.267	0.736

For the definition of γ see eq. (3) in Section 2.

We use a validation dataset for model selection and an independent dataset for testing, both containing 400,000 claims. To train our machine-learning models, we use the remaining part of the sample. Evaluation results are reported in Table 2. As we can see, the self-attention model performs best. We observe a relevant increase between the performance of convolutional neural networks (CNN), hand-designed features and boosted trees, bag-of-words (BOW), and our proposed methods based on position-wise feed-forward (PFF) and self-attention (SelfA). We explain this difference in performance by the fact that self-attention and position-wise feed-forward networks retain the relationships between variables from the same claim item, while these are lost when using all other methods. From these results, we infer that the interactions between variables from the same claim item are important in fraud detection systems. Furthermore, the self-attention model is able to form a context-dependent representation of the claim items, which explains the additional performance increase compared to the position-wise feed-forward model.

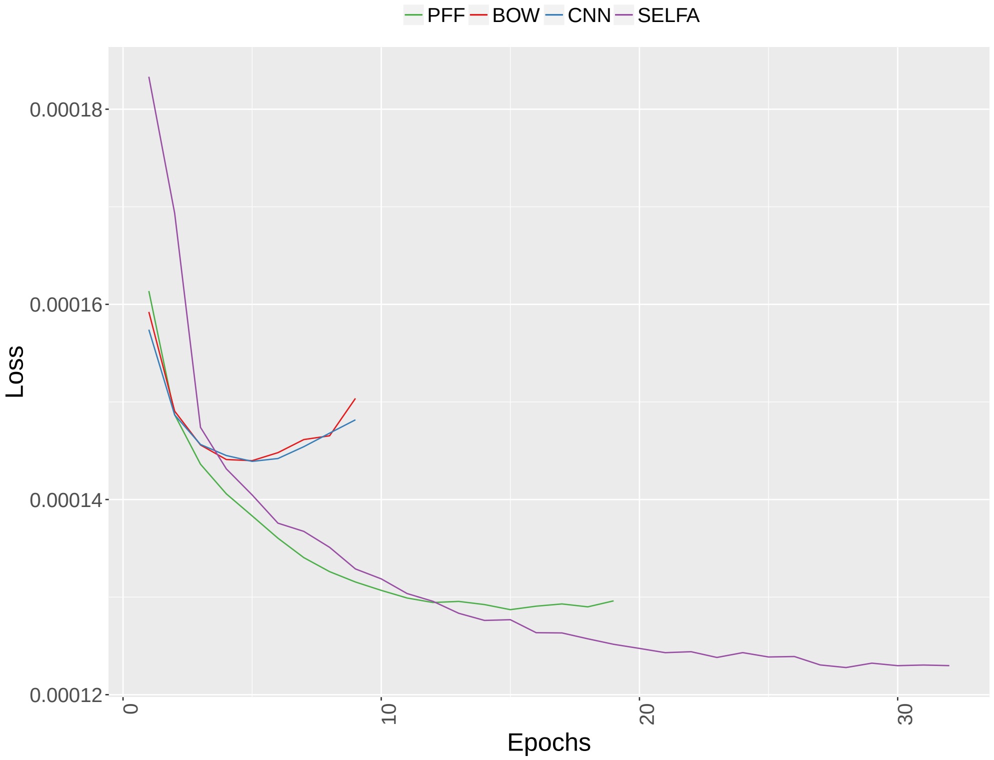
Moreover, in Figure 5 we can see that the CNN and BOW models quickly start to overfit, whereas the PFF and self-attention models reach much lower loss values, with self-attention again performing best. We use early stopping for all methods, which explains the difference in training epochs. The convergence plot is constructed from the validation data.

6. Fraud Mechanisms

Neural attention was a cornerstone in the development of many modern deep learning methods. In our application, it not only increases the quality of predictions but also adds, with the estimation of the attention distribution, a potentially informative tool to the decision-support system. The attention weights may provide meaningful explanations for predictions and enable further insights into the relative importance of the inputs – that is, the claim items in our application. In the position-wise feed-forward model, we use attention only for aggregation, whereas in the self-attention model we use it for both feature extraction and aggregation. For both models, we use the attention weights from the aggregation layer to analyze their meaning in the prediction task.

Importantly, the auditing unit benefits from attention weights only if these provide meaningful explanations, because only then can the weights shed light on which claim items are particularly

Figure 5 **Validation loss convergence plots of our different models.**



relevant for classifying a claim as being suspicious. Clearly, such information would speed up the subsequent (manual) auditing process of the suspicious claims. An open research question, however, is the extent to which the obtained attention weights deliver meaningful insights into the relative importance of the inputs. The results of recent studies suggest that the explanations provided by attention weights may not, in fact, be meaningful (Jain and Wallace 2019). In this section, we contribute to this strand of the literature in the context of health insurance claims management. For this purpose, we perform a simulation study that allows us to compare the estimated attention weights with the *known* causal reasons for a suspicious claim.

We generate artificial claims data as a sequence of two categorical variables per claim item. Each of the two variables contains 20 categories. We draw the categories from a uniform distribution. Each claim consists of up to 100 claim items. This results in a data structure similar to the claims data in our empirical application. Our simulated dataset contains 100,000 artificial claims. We label a claim as suspicious following one or multiple deterministic conditions. It is important for each condition to comprise multiple claim items and depend on a combination of the categorical variables as this is how we expect suspicious cases to occur in the real world. Because we artificially generate the claims data, we know the cause of each suspicious label and can therefore verify the extent to which our models are able to recover these sequence elements.

We use different sets of deterministic fraud conditions to generate invalid combinations of claim items. In the first setup, we generate a suspicious case if the following two conditions hold jointly: The claim contains an item with the constellation (19,1) and an item with the constellation (17,8). This way the models must capture both the interactions of variables from a single item and the interactions between different items. Note that this setup is an impossible task for the bag-of-words model because it loses the relationship between variables from the same claim item. In the other setups, we include multiples of the deterministic conditions discussed above with various combinations of categories – whereby the occurrence of one of these causes the claim to be suspicious in the simulated data.

We say that a prediction has been correctly explained if the model puts high (attention) weights on the correct invalid claim items. To evaluate the quality of the explanations in our simulations, we say that a claim item is the explanation of a suspicious prediction if the attention weight is larger than some threshold. We set the threshold at 0.5, but it turns out to be irrelevant as the attention weights become saturated at zero and one after some training time.

Obtaining meaningful explanations is a core task of prescriptive analytics. We use the following measures to evaluate the extent to which our deep learning models can explain their predictions:

1.) How many times did the model recover all relevant invalid items (variable screening)?

2.) And, subsequently, upon how many additional (valid) items did the model place weight? Here we take the average and maximum number of claim items flagged as suspicious. In case of perfect prediction, the average and maximum will be two.

After generating the data, we split it into training and test datasets. We train both a position-wise feed-forward and a self-attention model on the training data and assess the quality of the results in the test data. We first note that both models can easily achieve a perfect prediction performance, meaning that they can fully separate between fraudulent and non-fraudulent cases. If the machine-learning model cannot deliver meaningful attention weights in our simplified setting, there is little hope that it will deliver them in more realistic settings where systematic fraud or upcoding may be covered by noise.

In general, we find that the explanations of the position-wise feed-forward model are very good (see Table 3). This model places most of the attention mass on the two claim items that are the reasons for the invalidity of the claim. Moreover, the model reduces the number of relevant claim items considerably. Employing the information in the attention weights, we can reduce the number of problematic items to 2.88 on average in the setup with three conditions. The worst-case claim contains six potentially problematic items. This is a successful reduction of claim complexity considering that the average number of claim items is around 50. These results suggest that the

Table 3 Simulation results (Position-wise feed-forward model).

Conditions	Share	Screening	Average	Max
1	0.02	1	2	2
2	0.03	1	2.48	4
3	0.05	1	2.88	6

SHARE denotes the fraction of fraudulent claims.

position-wise feed-forward model delivers meaningful explanations. Exploiting these in the subsequent management process has great potential to speed up the manual auditing decision.

In contrast, the explanations derived from the self-attention model are not at all meaningful in our simulations. Even in the simplest case with just one deterministic condition and after intense hyperparameter variation, the model was not able to recover the invalid claim items via the attention weights. This result is in line with recent literature on text classification (Jain and Wallace 2019).

An interesting result is that the maximum number of marked items reflects the number of claim items that contribute to a potential fraud mechanism. Here we see a potential weakness of the position-wise feed-forward model: The representations of the inputs are context-independent and the attention function is also fixed. As a result, we will always mark a claim item as relevant if it is part of a potential fraud combination (even if the other item in this combination is not part of the claim). To illustrate, imagine we have two fraud conditions, either [(19,1) and (17,8)] or [(15,2) and (13,7)] in the same claim. Now, the cases where the model marks too many inputs as relevant are cases where, for example (19,1), (17,8) and (15,2), are part of the same claim. This highlights a potential advantage of the self-attention model: Because the input representations are context-dependent, it is theoretically possible to solve this problem perfectly. However, beforehand, further research is necessary to address the lack of explainability of the self-attention model. A combination of the attention weights from the feature extraction and aggregation layer may be a fruitful way to proceed in this matter.

7. Conclusion

We have described the important similarities between the way that texts and health insurance claims are structured, and how these allow deep learning techniques from text classification to be extended to the task of classifying claims. Claims management is a promising application for these methods outside of natural language processing or image analysis, and has real and quantifiable economic value. Methods based on neural networks can improve upon or even replace hand-designed features.

In this study, we use deep neural nets to set up a fraud detection model for claims data, which not only classifies claims but also delivers meaningful explanations. We propose a machine-learning architecture that is tailor-made for the structure of claims data. It may also be applicable, in a wider context, to applications with similar hierarchical data structures. Our proposed architectures outperform default models on a dataset comprising two million claims from a private insurance company in Germany. A comprehensive data-driven model of claims management can be built in a straightforward manner by incorporating additional input variables, such as information about health care providers or variable length information.

Our empirical results substantiate the performance of our models as data-driven tools in predictive analytics. Moreover, we conduct a simulation study to assess their utility as a tool in prescriptive analytics. Our simulations provide the first evidence that one of our models is able to deliver meaningful explanations of its predictions. These explanations can be used to mitigate informational asymmetries and accelerate subsequent auditing decisions.

References

Ba JL, Kiros JR, Hinton GE (2016) Layer Normalization. *arXiv preprint arXiv:1607.06450* .

Bastani H, Goh J, Bayati M (2018) Evidence of upcoding in pay-for-performance programs. *Management Science,* forthcoming.

Becker D, Kessler D, McClellan M (2005) Detecting medicare abuse. *Journal of Health Economics* 24(1):189–210.

Bolton RJ, Hand DJ (2002) Statistical fraud detection: A review. *Statistical science* 17(3):235–249.

Bond EW, Crocker KJ (1997) Hardball and the soft touch: the economics of optimal insurance contracts with costly state verification and endogenous monitoring costs. *Journal of Public Economics* 63(2):239–264.

Breiman L (2001) Random forests. *Machine learning* 45(1):5–32.

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) *Classification and Regression Trees* (Wadsworth), ISBN 0-534-98053-8.

Cecchini M, Aytug H, Koehler GJ, Pathak P (2010) Detecting management fraud in public companies. *Management Science* 56(7):1146–1160.

Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .

Dionne G, Giuliano F, Picard P (2009) Optimal auditing with scoring: Theory and application to insurance fraud. *Management Science* 55(1):58–70.

Ekin T, Ieva F, Ruggeri F, Soyer R (2018) Statistical medical fraud assessment: exposition to an emerging field. *International Statistical Review* 86(3):379–402.

Fang H, Gong Q (2017) Detecting potential overbilling in medicare reimbursement via hours worked. *American Economic Review* 107(2):562–91.

Graves A (2012) *Supervised Sequence Labelling with Recurrent Neural Networks* (Berlin, Heidelberg: Springer).

Guo C, Berkhahn F (2016) Entity Embeddings of Categorical Variables. *arXiv preprint arXiv:1604.06737* .

Heese J (2018) The role of overbilling in hospitals earnings management decisions. *European Accounting Review* 27(5):875–900.

Jain S, Wallace BC (2019) Attention is not explanation. *arXiv preprint arXiv:1902.10186* .

Kim Y (2014) Convolutional Neural Networks for Sentence Classification. *arXiv preprint arXiv:1408.5882* .

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. *Twenty-ninth AAAI conference on artificial intelligence*.

Li J, Huang KY, Jin J, Shi J (2008) A survey on statistical methods for health care fraud detection. *Health care management science* 11(3):275–287.

Liou FM, Tang YC, Chen JY (2008) Detecting hospital fraud and claim abuse through diabetic outpatient services. *Health care management science* 11(4):353–358.

Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781* .

Mookherjee D, Png I (1989) Optimal auditing, insurance, and redistribution. *The Quarterly Journal of Economics* 104(2):399–415.

Picard P (1996) Auditing claims in the insurance market with fraud: The credibility issue. *Journal of Public Economics* 63(1):27–56.

Picard P (2013) Economic analysis of insurance fraud. *Handbook of Insurance*, 349–395 (Springer).

Schiller J (2006) The impact of insurance fraud detection systems. *Journal of Risk and Insurance* 73(3):421–438.

Shen T, Zhou T, Long G, Jiang J, Pan S, Zhang C (2017) DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding. *arXiv preprint arXiv:1709.04696* .

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

Townsend RM (1979) Optimal contracts and competitive markets with costly state verification. *Journal of Economic theory* 21(2):265–293.

Urbanovich E, Young EE, Puterman ML, Fattedad SO (2003) Early detection of high-risk claims at the workers' compensation board of british columbia. *Interfaces* 33(4):15–26.

Van Vlasselaer V, Eliassi-Rad T, Akoglu L, Snoeck M, Baesens B (2017) Gotcha! network-based fraud detection for social security fraud. *Management Science* 63(9):3090–3110.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. *Advances in Neural Information Processing Systems*, 5998–6008.

Viaene S, Derrig RA, Baesens B, Dedene G (2002) A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection. *Journal of Risk and Insurance* 69(3):373–421.

Wager S, Wang S, Liang PS (2013) Dropout training as adaptive regularization. *Advances in neural information processing systems*, 351–359.

Wu S, Zhong S, Liu Y (2018) Deep residual learning for image steganalysis. *Multimedia tools and applications* 77(9):10437–10453.