Ryan Filgas
AI Grad Research Paper

Explaining Anomalies using Denoising Autoencoders for Financial Tabular Data is a paper written by Timur Sattarov, Fayananda Herurkar, and Jorn Hees about explainable AI with the intention of providing domain experts in financial data with a framework to understand abnormal characteristics of data anomalies and improve in-house data quality management.

The goal of the research in this paper is to explain which features contribute to high anomaly scores and assist a reporting agent in making corrections. The paper proposes using a denoising auto encoder (DAE) neural network that isolates anomalous data points and flags the fields that cause the irregularity. These are focused used for financial tabular data with categorical and numerical type using flagging method for individual cells rather than records in addition to estimating expected value. The authors present three contributions: denoising autoencoder neural networks can explain causes of irregularity of a sample for mixed tabular data, a model can successfully detect reporting errors on the cell level providing corresponding confidence scores and estimated for fixing them; they also propose an extension with enhanced loss (Sattarov, 1-2).

Past autoencoder techniques were used to detect money laundering and fraud or learn behavioral fraud features. Some techniques used in XAI include Shapley Additive exPlanations (SHAP), Local Interpretable Model Agnostic Explanations (LIME) and DeepLIFT. These models can explain almost any machine learning model (Sattarov, 2).

The auto encoder neural network denotes a set of instances in a tabular dataset. Each instance encompasses a set of attributes with either a numeric real number or a categorical type. The autoencoder is a feed forward NN that performs data compression into a lower dimensional feature space before reconstructing it with minimal loss. The reconstruction error reflects how well an instance fits into the general patterns of the data. This difference means that a datapoint that fits well into the data has low reconstruction loss. Which the network is trained to minimize. (Sattarov, 3).

Because the data is mixed, reconstruction loss is a combination sum of two losses: the negative-log-likelihood loss is used for categorical attributes, and the mean squared loss for numerical attributes. A DAE is trained by disrupting input data with noise and reconstructing the clean data. In addition to removing noise, it improves hidden layer robustness and reduces overfitting risk. To address the tradeoff between noise removal and anomaly detection, researchers introduce an alpha parameter to weigh noise removal vs cleaning data. To explain anomalies researchers utilized reconstruction error of separate attributes to address which samples are anomalies, why they are anomalies, and what should be reported instead. While training, the DAE reconstructs a noiseless instance from its corrupted counterpart while minimizing loss between the two (Sattarov, 3).

Researchers used the pytortch framework for testing using open source and real-world datasets. These included the credit default dataset from the UCI ml repository, IEEE Fraud from the ecommerce Vesta corporation, Adult from the UCI ml repository, and Holdings a set of proprietary individual holdings of investment funds. To test researchers artificially generated anomalies to turn 3% into outliers. 50% of the features were corrupted uniformly at random. The injection of noise uses an additive process where the noise is sampled from Gaussian, Laplace, or Log-Normal distributions. For categorical features two alternatives are used, the first where the original entry is replaced by picking a categorical entry at random from the distinct values of the attribute. The second option performs character manipulation ensuring a new entry is created (Sattarov, 4). To evaluate the technique proposed they used three metrics: Precision at K – “which samples are anomalies?”, mean average precision – “why is it an anomaly?”, and mean expected value – “what should have been reported instead?” The model training setup is split 70/30 for train/test. Each model is trained for 5000 epochs with a minibatch size of 128 and an adam optimizer with B1 = .9 and B2 = .999 with a cosine learning rate scheduler. The parameters of the encoder and decoder are randomly initiated. The technique is compared with PCA, Marginal Distribution, and a traditional auto encoder (Sattarov, 5).

The researchers found positive results. The domain expert is provided with a visual inspection tool where each potential anomaly can be screened through flagging individual cells for detected errors. For each case an estimated value for what should have been in that cell is provided, and five similar data points are given for comparison. This framework provides more explanation capabilities, saves screening time, and reduces human error during quality checks (Sattarov, 6).

DAE seems to group similar datapoints in a more compact way vs the sparse distribution of the AE. This makes DAE much more valuable to a domain expert who wants more clear group separation. In addition DAE out performs baseline AE’s on almost all metrics and datasets tested. This is largely due to DAE containing smaller reconstruction error. Researchers also found that an autoencoder trained on clean data normally does better than those trained on anomalies. This indicates that it’s better to deploy an AE model trained on clean data. Overall researchers were able to put together a successful framework to be used as a tool for data quality domain experts to detect and analyze anomalies in their data (Sattarov,6). On a reflective note, the researchers were very thorough in explaining their methods. They mention their methods could be expanded outside the field of financial data and it would be interesting and useful to test that and describe what some potential use areas might be.

Ryan Filgas
AI Grad Research Paper

Sources Cited

Sattarov, T., Herurkar, D., & Hees, J.. (2022). Explaining Anomalies using Denoising Autoencoders for Financial Tabular Data.