British Actuarial Journal (2026), Vol. **31**, e13, pp. 1–15
doi:10.1017/S1357321726100415

Institute and Faculty of Actuaries

SESSIONAL MEETING DISCUSSION

Hush hush: Keeping neural networks claims modelling private, secret, and distributed using federated learning: Discussion

[IFoA Sessional Webinar, Thursday 8 May 2025]

This discussion paper relates to the paper presented by Dylan Liew, Dr Małgorzata Śmietanka, Harry Loh, Michelle Chen and Scott Hand at the IFoA sessional webinar held on 8 May 2025.

Moderator (Mr A. E. Iglauer, F.I.A.): Welcome to our session today. My name is Alexis Iglauer. I chair the research section of the Data Science group within the Institute and Faculty of Actuaries (IFoA). The work presented today is by one of the working parties within the research group looking specifically at Federated Learning.

I find this paper particularly interesting because it highlights how actuaries can leverage data science. Federated Learning is a very well-developed area that the actuarial field has not been using extensively until now. In addition, this paper takes techniques from other areas of data science and pulls them together in a way that cleverly allows multiple groups to fit models together without having to share the underlying data, which is a practical problem.

Mr D. I. Liew, F.I.A.: My name is Dylan Liew. I am the Head of Technical Pricing and Pricing Data at Bupa. A lot of modelling work involves cleaning and formatting data. We often expect modelling work to comprise of roughly 80% of time spent on data and 20% building the model. The amount of time we spend on data, gathering and cleaning will likely increase with time because of stringent data protection rules like GDPR. But in this age of AI this is creating counteracting forces. On one hand we are trying to build bigger and more data-hungry AI models; while on the other hand we are also trying to protect our data more. The rest of this session focuses on Federated Learning, which could provide a potential solution to this problem.

Let's begin with a well-known, relatable problem – the Netflix prize problem. In the mid-noughties, Netflix had the idea to outsource their movie-liking prediction model. They took a lot of customer data and released it to the public, anonymised. The goal was to try to get the lowest root mean square error on the data set. The team with the best model at the end would win $1,000,000. The data comprised of a movie ID, a customer ID, and a date. They were ultimately trying to predict what the customer's rating of that movie was.

Netflix was challenged on whether they were allowed to release this customer data, and if they had obtained consent from customers to publicly show their movie ratings? Netflix responded that these are movie ratings, not private banking information or health records, and also that they were anonymised. However, it was observed that when people rated a movie on Netflix in a certain way, they often later went on IMDb and rated the same movie, the same way, on the same date. Some clever researchers worked out that they could find a user in the Netflix prize data set and match that same real person on IMDb, which led to the so-called robust de-anonymisation of the Netflix prize problem.

© The Institute and Faculty of Actuaries, 2026. Published by Cambridge University Press on behalf of The Institute and Faculty of Actuaries. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution, and reproduction in any medium, provided the original work is properly cited.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

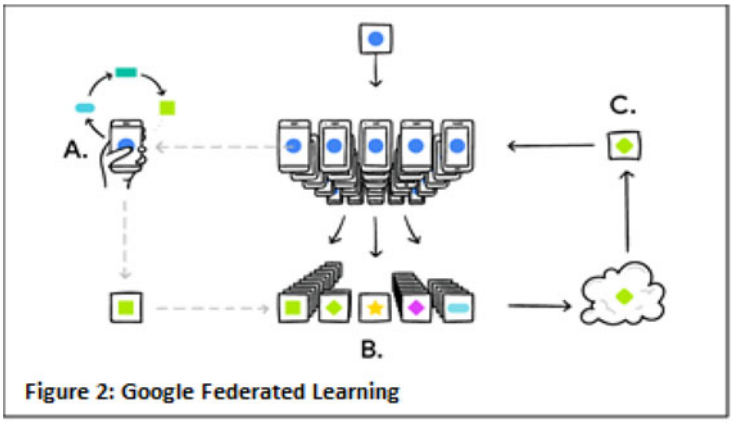
The authors went on to say that, whilst Netflix said this was "only" movie ratings, perhaps movie ratings say something about them that could be confidential, like their religious or political views, BMI, or food preferences.

A US demographics study (Sweeney, 2000) had worked out that around 87% of the US population can be uniquely identified by just their age, sex and zip code alone. Sometime later the Massachusetts Group Insurance Commission (GIC), a quasi-state-owned health insurer, decided to anonymise their employee health records and release them to the public in much the same way Netflix did, to allow researchers to study that data set, build predictive models, and find some value and insights that could help people.

However, Sweeney had warned the project sponsor, William Weld, and said that even though names were removed it was confidential information. Sweeney purchased, for about $20, data of all voters from a local library, to find out William Weld's own zip code, gender, date of birth, and found his own health records in that data set and then mailed them back to him to prove the point. Later, Sweeney managed to de-anonymise the human genome project, which is probably the most personal information one can possibly get, because this describes individuals down to a genetic level.

A more recent example of this type of attack was the "Strava Leaks". In 2018, the US Pentagon realised that military personnel were posting their Strava runs late at night, which led to a lot of security breaches. While leaders do not often post their Strava runs, their bodyguards or the friends and families of bodyguards do. These are all examples of linkage attacks, which make it very difficult to keep data private.

Dr M. Śmietanka: Let's now discuss how Federated Learning works. The most common use-case of Federated Learning in the context of mobile phones is prediction in text messages. In traditional Machine Learning, all the data is collected and the model is trained on that aggregated data, but there are a few problems with that approach. Firstly, it takes a lot of time and memory resources, but it also requires consent because of the privacy of the data.



A) your phone personalises the model locally depending on your usage;

B) many users' updates are aggregated;

C) the aggregated updates form a consensus change to the shared model; and

D) the shared models are updated.

Figure 1. Smartphone federated learning pipeline.

To tackle this issue, Google started with a randomly calibrated model and left it to users to improve it. Figure 1 illustrates how this model works. When you plug in your mobile and are connected to Wi-Fi, every now and then there is a little bit of training on your local data on your mobile phone. Once the model is trained your device sends model parameter updates to the Cloud, where it is aggregated with everyone else's updates in the Cloud that took part in that round of training. The centrally aggregated and updated average global model parameters are then broadcast back to all users. The beauty of the approach is that you benefit from the smarter model, without having to share your underlying personal data and have it centrally collated.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

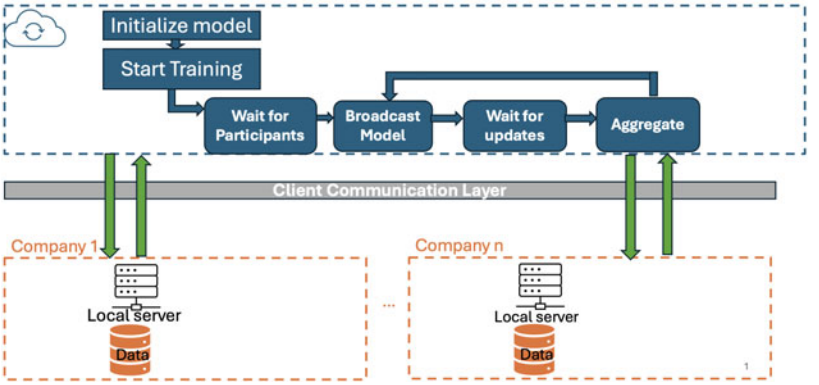


Figure 2. Insurance federated learning pipeline.

To put this in the context of insurance, let's consider a couple of companies wanting to collaborate on a Machine Learning model to predict claims frequency. Figure 2 shows a schematic of a Federated Learning pipeline for insurance. Companies in the network can agree on a set of parameters, a set of hyper-parameters, a loss function and the model architecture. Then the companies choose the central body, who will be acting as an aggregator. It will collect all the model parameter updates and will update the global model. It will also broadcast the model parameter updates to all the participants. The training loop is the same as in the use-case of mobile phones. As a result of this Federated Learning model training, each insurer gets a smarter model, even though they do not generate enough data for that model or share their sensitive customer data. The only information shared with the aggregator and with everyone in the training loop is model parameter updates, not data.

Mr Liew: There are still some flaws and problems in what we just described. Consider that, whilst we only share model parameters, perhaps insurance companies use claims frequency models, such as GLMs or neural networks. They send these parameters (let us call them beta) to the aggregator. The aggregator could be a bad actor, or that parameter could be intercepted along the way. The issue is that it is not impossible to infer something from the parameter, such as a higher beta for an insurer means the higher their claims experience.

Therefore, the problem we have is, we need to work out a way to average a set of numbers without exposing those numbers to an aggregated body. For example, if we consider three different insurers- insurer0, insurer1 and insurer2 – with parameters beta0, beta1, beta2 respectively, the problem is how do they collect that information together in a secure way? To keep things simple let's just set the value of beta0 to zero.

Before we do the aggregation, the insurers are going to "pair-wise exchange" random numbers, as Figure 3 illustrates. For example, say insurer0 generates the number 17 with insurer2. When insurer2 receives that random number 17, insurer2 cannot infer anything about beta0. They do not know if beta0 is more than 17 or less than 17. All they know is that insurer0 gave them a random number. Insurer0 could also send another random number, say this time they generate 2.3, and send that to Insurer1. Again, Insurer1 is going to receive a 2.3 and not know what to infer from it. We repeat the process and all of the insurers 'pair-wise exchange' these random numbers. Each insurer is going to have sent out two random numbers, and they have received two random numbers from the other insurers.

For insurer0, they sent 17 and 2.3 and received –99 and 5. Then, in private, on insurer0's own servers they add the value of the masks, or the random numbers they sent out, and deduct the value of the masks that they received. In this case, you would get about –84.7.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

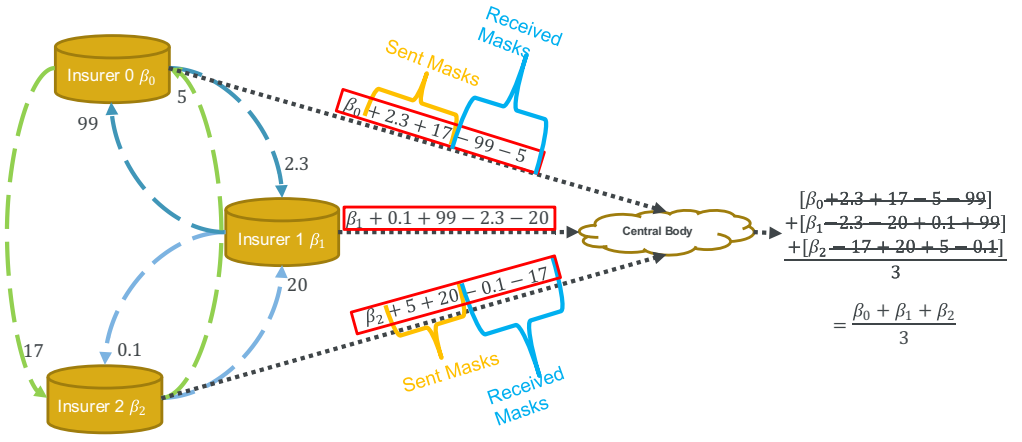


Figure 3. Need to encrypt parameters.

They are then going to send that –84.7 to the aggregating centralised body, which could be a statutory body, a regulator, an auditor, or a re-insurer. Notice how the aggregating body, when they receive –84.7, are also none-the-wiser as to what beta0 really is. They do not know how the parameter compares with –84.7. The other two insurers will do the same thing. The aggregating body will receive, essentially, three garbled or encrypted numbers, and they will just add those together. If we look at the maths, you can see behind the scenes what is happening. All those random masks that the insurers have exchanged with each other, because they are adding them on and deducting them, will cancel out at the point of aggregation. And just like that, the aggregating body has produced the average of the three parameters without anyone, at any point, being exposed to the true parameters other than the insurers themselves.

Ms M. Chen: Thanks, Dylan (Liew). I am Michelle (Chen), a Capital Actuary at Tokio Marine Kiln. Next, we are going to discuss applications of the Federated Learning concept. The first application could be reinsurance, as shown in Figure 4, with reinsurance pricing determined from pooling data from insurers.

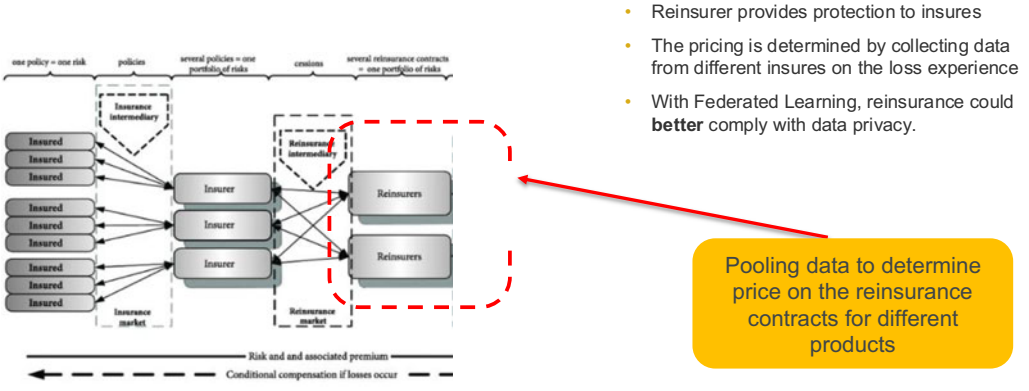


Figure 4. Application to reinsurance.

In general, the reinsurer has less data compared to direct insurers because they go through a different process with a reporting delay. For reinsurance to achieve the most appropriate price, they pull similar risks together, analyse the loss trends and train their own model. This way of collecting data, training your own model and updating parameters is like Federated Learning.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

With Federated Learning you could better comply with GDPR and data privacy issues. In the next section, Harry (Loh) will talk more about other applications.

Mr H. Loh: I am Harry (Loh), a GI Actuary in a consultancy in Singapore. I previously worked at a syndicate in London, in an AI risk role, within Lloyd's. Lloyd's of London is a very old insurance body, probably one of the oldest in the world, so that means it has a lot of data. The problem is that the data is scattered across all the different syndicates that exist within Lloyd's of London. How can all of these different syndicates collaborate securely to use the internal data sets to build better pricing models? Federated Learning can be one of the ways. In the London market, we write some very low-frequency, high-severity products, like space shuttle insurance or event cancellations - products that have very few insurance claims. It is in the interest of the syndicates to work together, to get more accurate pricing models. Since they work across borders, it might be easier to share data across different branches in the world using Federated Learning.

We now move on to a Health insurance use case involving wearables. There have been many ways to anonymise the data by removing names or other parameters, but sometimes that is not enough because it may be possible to identify a specific person with just three parameters. Federated Learning is one of the ways that can help many different Health insurers build better models while keeping the customer data private.

Federated Learning can be used to collect health data from wearable devices. This means that the private health data of every individual remains on each device and is not transmitted to a centralised data server. Each device would then train a Machine Learning model, and the aggregate insights of local models are used to train a Global Model. Everyone that contributes to that Global Model benefits from better predictions.

Another real application that is backed by Nvidia is to be implemented in the US. They are working across different hospitals and medical research facilities to build Machine Learning models to better diagnose cancerous brain tumours. By gathering a lot of data to build the Machine Learning models, medical consultants gain additional points of accuracy, which makes it close to what a specialist medical consultant can obtain over many years of experience.

One final example of how Federated Learning has been used is in fraud detection and is backed by the company Swift. Swift is a huge payment provider through the world, working with many different companies that use their payment network. They are building a sandbox along with 12 other financial institutions with synthetic data and using Federated Learning to build better models to combat financial fraud.

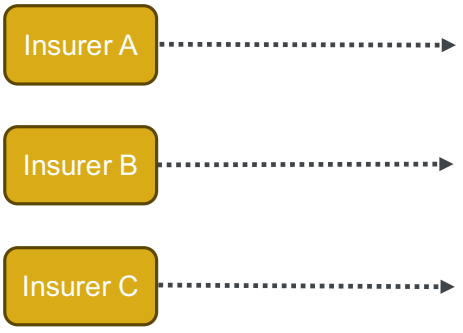
Mr S. Hand: Hi everyone. My name is Scott (Hand). I have worked as an actuarial analyst in the Health and Life space for two and a half years now. As many of you are probably already aware, in the Life and Health industry it is common practice to set mortality and morbidity assumptions relative to standard tables (Figure 5). The way these standard tables are calculated is that many insurers in the protection market send their anonymised data to the CMI, who are part of the IFoA. The advantage of this is having more data, which leads to more accurate mortality rates by age, where there is less data available on each sub-group. Federated Learning could be an alternative to this approach.

Ms Chen: Thanks Scott (Hand). We will next look at a sample case in insurance claims by the Federated Learning Working Party. The French motor third-party liability claims data set (publicly available) collates data from 700,000 policies over the year.

Table 1 (in Figure 6) lists the fields that represent different columns (variables) in this data. This data has been widely used in actuarial research. We uniformly and randomly split the dataset into ten different data sets, each with the same number of rows, and each one representing different insurers (Figure 7).

With each data set we then use neural networks as our model. This works like a GLM. Weights and biases of the neural network can be thought of as coefficients in a GLM. The activation function is equivalent to the link function.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press



Age x	Curtate duration t					
	0	1	2	3	4	5+
18	0.000153	0.000194	0.000209	0.000218	0.000218	0.000241
19	0.000153	0.000194	0.000209	0.000218	0.000218	0.000241
20	0.000153	0.000194	0.000210	0.000219	0.000219	0.000242
21	0.000153	0.000194	0.000210	0.000219	0.000219	0.000243
22	0.000153	0.000194	0.000211	0.000220	0.000220	0.000244
23	0.000153	0.000195	0.000211	0.000221	0.000221	0.000246
24	0.000153	0.000195	0.000212	0.000222	0.000222	0.000248
25	0.000153	0.000196	0.000214	0.000224	0.000224	0.000250
26	0.000153	0.000197	0.000215	0.000225	0.000225	0.000252
27	0.000153	0.000199	0.000217	0.000227	0.000227	0.000255
28	0.000153	0.000200	0.000219	0.000230	0.000230	0.000258
29	0.000154	0.000202	0.000221	0.000232	0.000232	0.000261
30	0.000155	0.000204	0.000224	0.000235	0.000235	0.000265
31	0.000155	0.000206	0.000227	0.000238	0.000238	0.000269
32	0.000156	0.000209	0.000230	0.000242	0.000242	0.000273
33	0.000157	0.000211	0.000233	0.000246	0.000246	0.000278
34	0.000160	0.000217	0.000239	0.000252	0.000252	0.000286

Figure 5. CMI application.

- The **freMTPL2freq** car insurance claims dataset – Publicly available
- 677,991 motor third-party liability policies (observed on a year)

Table 1. Description of data, fields, and preprocessing transformations used in experiment

Field	Description	Transformation
IDpol	Unique policy number	Dropped
ClaimNb	Number of claims on the given policy	Capped at 4
Exposure	Total exposure in yearly units	Capped at 1
Area	France area code (categorical, ordinal)	Ordinally encoded e.g. $A:1, B:2, C:3$ etc.
VehPower	Horse power of the car (categorical, ordinal)	*MinMaxScaler*
VehAge	Age of the car in years	*MinMaxScaler*
DrivAge	Age of the driver in years	*MinMaxScaler*
BonusMalus	Bonus-malus (i.e. No Claims Discount) level between 50 - 230	*MinMaxScaler* after capping at 150
VehBrand	Car brand (categorical, nominal)	One-hot-encoded
VehGas	Diesel or petrol car (binary)	Ordinally encoded i.e. $Regular:1, Diesel:2$
Density	Density of inhabitants per km2 in the city of the residential address of the driver	*MinMaxScaler* after $\log$ transforming
Region	Regions in France prior to 2016 (categorical)	One-hot-encoded

Figure 6. Data on French motor claims.

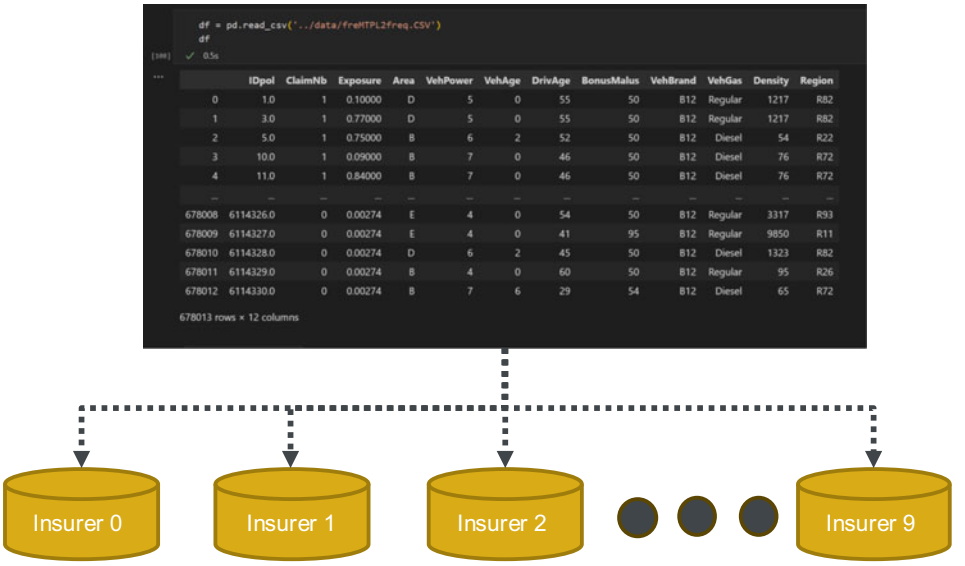


Figure 7. Insurance federated learning use case.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

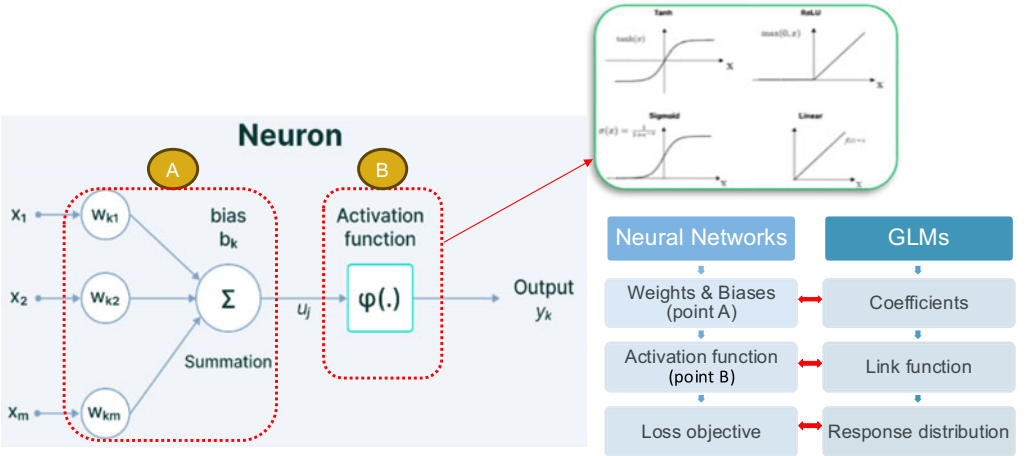


Figure 8. Neural networks.

Section A in Figure 8 has the inputs variables, such as vehicle age and region. These inputs are applied to the weights and then biases. This then goes to the activation function. The activation function is important to a neural network, as it introduces the non-linearity that allows the neural network to capture more complex relationships between inputs and outputs. Finally, the loss objective in the neural network measures how well your model predicts the output compared with the real data and is similar to the response distribution in GLMs.

Table 2. Neural Network Architecture used in all 3 Scenarios

Hyperparameter	Selection
Input neurons	39 based on the preprocessing done in Section 5.2.2
Hidden Layers	2
Output Layer	1 output neuron with exponential link function (to ensure only positive frequencies are predicted)
Optimiser	*NAdam*
Activation Function	*tanh*
Loss Function	Negative Poisson Log Likelihood
Initialisation	Xavier
Epochs	300

Table 3. Hyperparameter Search Space Considered in all 3 Scenarios

Hyperparameter	Search Space
Learning Rate	[0.001, 0.002, 0.01]
Number neurons in Hidden Layer 1	[5, 10, 15, 20]
Number neurons in Hidden Layer 2	[5, 10, 15, 20]
Batch Size	[500, 1,000, 5,000, 10,000]

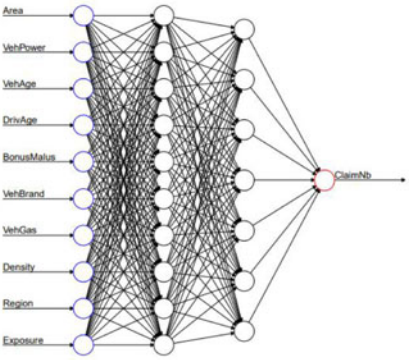


Figure 9. Neural Network model set up.

In our study, we used the NAdam optimiser and tanh() activation function (details in Tables 2 and 3 of Figure 9).

Mr Hand: I am going to talk about the three different scenarios in our use case. The first scenario is the Global Model Scenario (Figure 10), in which the central data store has all the data from all the different insurers. This is never going to be the case, because each insurer will want to keep their data. Each insurer's data is sensitive and will most likely not be shared with other insurers.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

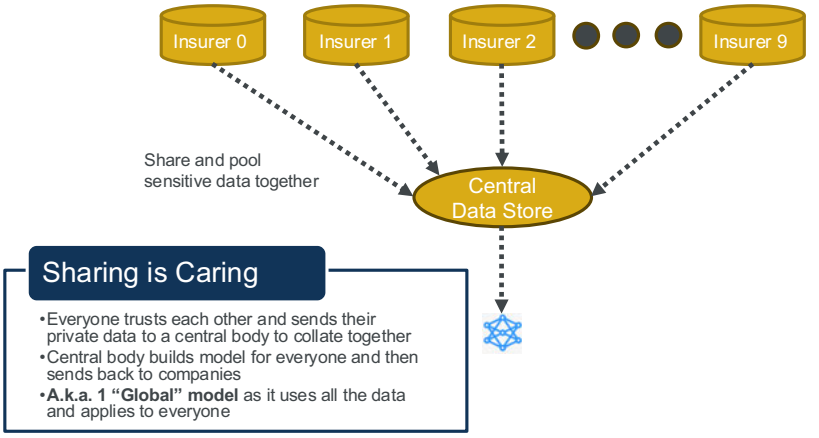


Figure 10. Global Model Scenario – 10 insurers, 1 model.

The second scenario is the Partial Model Scenario (Figure 11), when each insurer builds their own pricing model based on its own unique data.

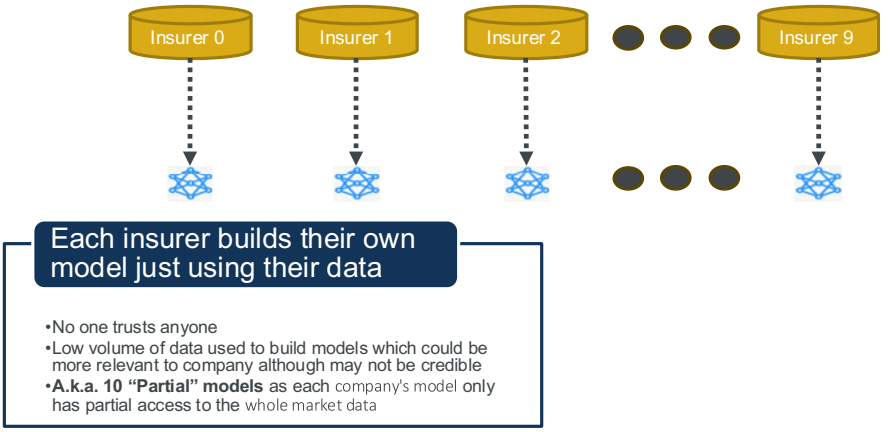


Figure 11. Partial Model Scenario – 10 insurers, 1 model.

Finally, we have the Federated Model Scenario (Figure 12) when each of the insurers is sharing the model parameters but not their own data. Their own data is kept on their own systems. The central body is controlling this, but it is important to highlight, as already discussed, that the central body cannot see these model parameters and only aggregates them together.

Mr Liew: To assess how all the models performed together, we took a randomly unseen test set out of the data and compared performance on that same data set. We measured model performance by the percentage of Poisson deviance explained, which is like an R squared, in the sense that the bigger the value the better the model is.

Figure 13 shows the comparison of results for the ten insurers. We see a spread of model results (Local Agents, green, bottom right), and between 2.8% and 3.8% of Poisson deviance explained with about a medium of 3.2%, even though they are all measured on the same data set. The Global Model (blue, top left) shows 5.57%. This is approximately twice the model performance of the local models acting independently.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

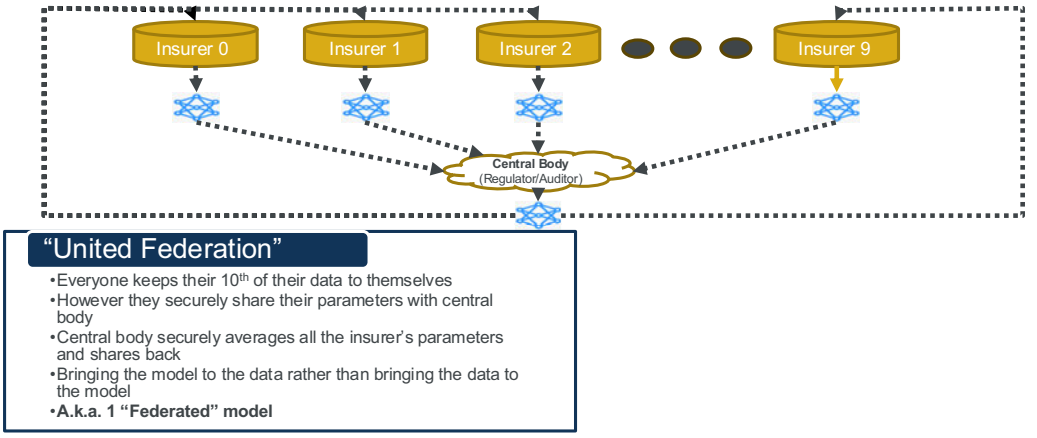


Figure 12. Federated Model Scenario – 10 insurers, 1 model.

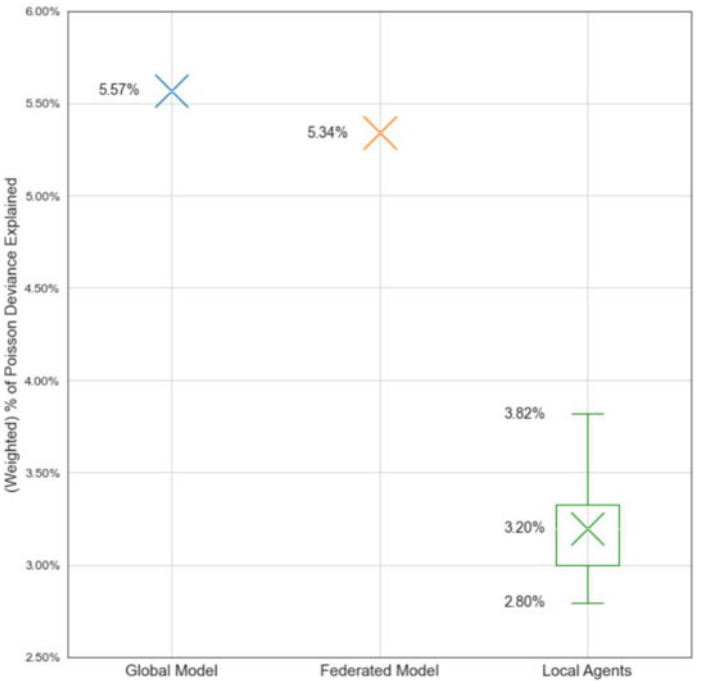


Figure 13. Comparison of results.

The Federated Model (orange, middle) got 5.34% on the test set. The Federated Model is not quite as good as the Global Model, but that is to be expected. The Federated Model is typically only used when a Global Model is not possible. But it comes close and is nearly twice as predictive as the local agents.

The bars in Figure 14 represent buckets or groups of customers and the lines represent number of claims experienced per group of customers. The Global Model's performance for that bucket of customers (very far end orange dot, value 1.25) has over-predicted the number of claims for that bucket of customers by about 25%. At the other end, the Global Model is under-predicting (orange dot, bottom right, value 0.7–0.8). The Federated Model (green) is also under-predicting in some

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

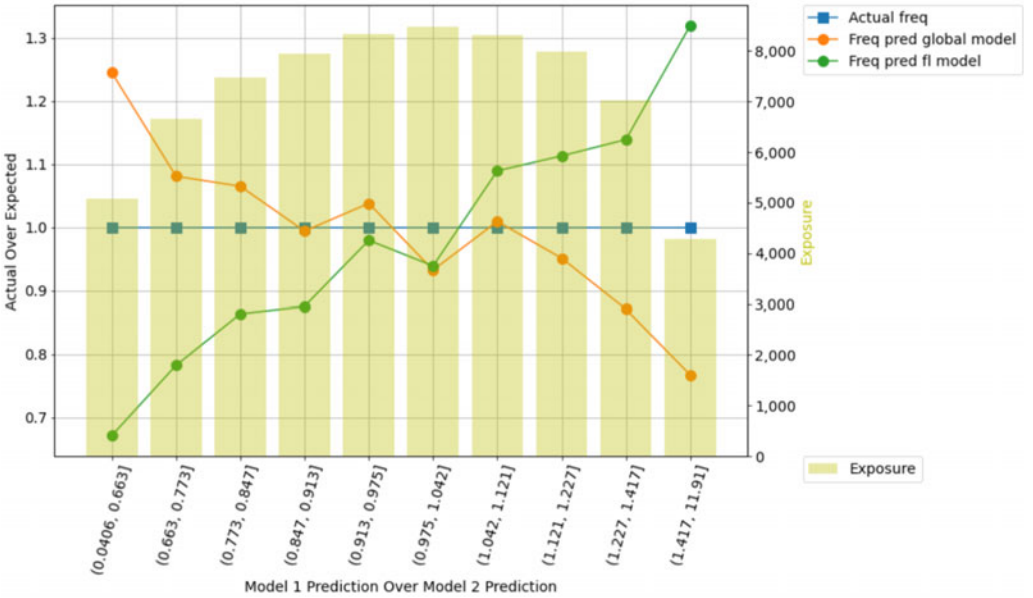


Figure 14. Further comparison of results.

regions and over-predicting in others. It is hard to tell which model is performing better because both models are over- and under-predicting by similar but opposite amounts in different regions.

Revisiting our example with 10 insurers with spread between 2.8 and 3.8 of Poisson deviance explained, the best insurer (the one that got the highest score) was in our case insurer number 5

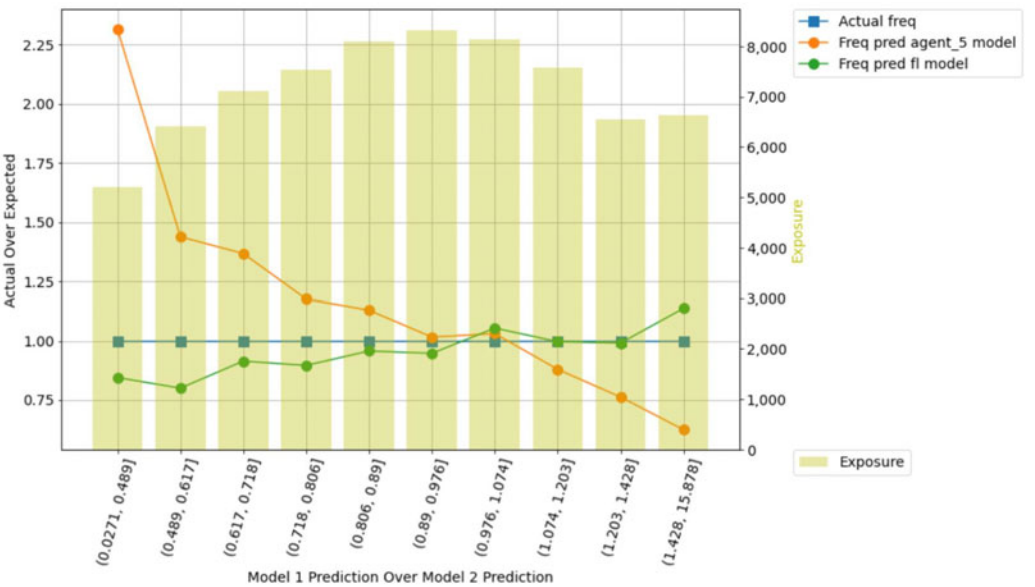


Figure 15. Comparison of insurer results.

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

(Figure 15). If we do a double lift chart, comparing the Federated Model against agent number 5, you can see it is a very different shape.

On the left-hand side of Figure 15, the global model overfits by almost 200%, hitting 2.25 and massively over-predicting the number of claims. The Federated Model (green line) is much closer to the actual claims. There is no cross-over (x-shape) as in Figure 14, making it easier to conclude that the Federated Model is better in this case.

Ms Chen: The Federated Learning seems to have a perfect solution for sharing the data. However, there are still some limitations (Figure 16) in Federated Learning in terms of data, features, scale, and encoding.

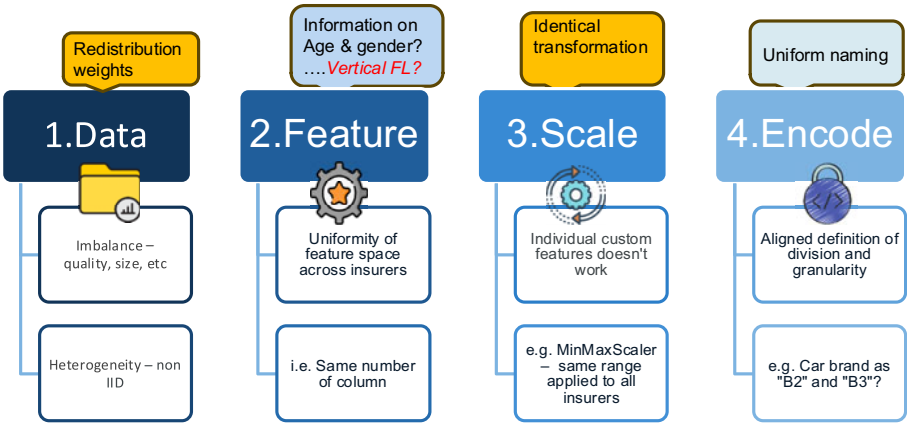


Figure 16. Limitations.

It could be imbalance between insurers, in terms of size or quality of the data. Large/leading insurers in the industry would be overrepresented in a Federated Learning exercise. Secondly, in terms of a features, this version of Federated Learning currently requires that all the participants need to have the same columns of data. In our application all ten insurers need to apply the same scale and transformations such as Min–Max scaling, which means they need to collectively determine factors, such as the minimum age of the drivers, which they could consider to be private.

For categorical variables, each insurer would also have to have the same encoding. This would mean insurers would need to also agree on groupings and granularity, say for instance in terms of the car brands.

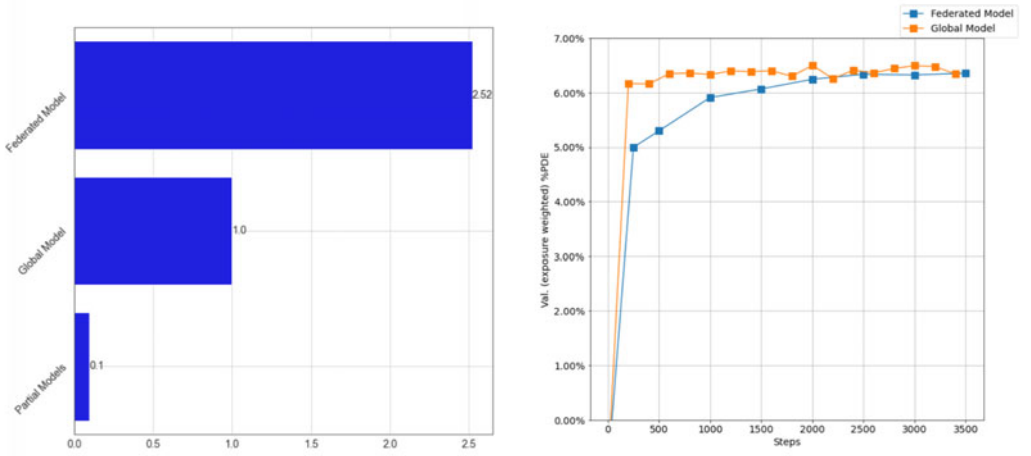
Dr Śmietanka: Our study finds that Federated Learning model can achieve very similar performance to a Global Model trained on aggregated data. We compared the time that it took to train a Global Model, ten Local Models, and a Federated Learning model.

Figure 17 shows that the Global Model took ten times longer to train compared to the Local Model. This is something that we expect because for Local Models we trained them on 10% of the data.

The Federated Learning Model takes 2.5 times longer to train compared to the Global Model. And why is that? This is not a like-for-like comparison. The Federated Learning model took much longer to converge. Chart (b) in Figure 17 shows that Federated Learning takes more iterations and more model parameter update steps to converge and hence the increase in training time.

What we did here as a Working Party was an application of Horizontal Federated Learning. All of the collaborators shared the same features about the claims. There are also other types of Federated Learning. Vertical Federated Learning is where collaborators share the same ID space. For example, they share the same customers, but they have different features. You can imagine a

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press



(a) Relative increase in observed wall time to train the models compared to training the Global Model.

(b) Exposure Weighted Validation % PDE of the Global and Federated Models over different number of parameter update steps.

Figure 17. What's the catch?

Health insurance company collaborating with a health provider to build a model to calculate better risk, for example for Type 2 diabetes.

Federated Learning has challenges beyond the data, on the technical side. There are system and operational challenges. Apart from longer training and convergence times, we must be aware of fault tolerance and client dropout. What happens if one of the agents participating in model training drops out? How do we protect the model from not affecting the Global Model results? There is also an adoption barrier - integration complexity with current systems as well as cultural resilience to adopt new methods, and potentially a skill gap if teams lack experienced people working with distributed data sets.

Mr Hand: One question that comes up is why have we not used GBMs, given that they typically have better model performance when working with actuarial data? Federated Learning has only been around since 2016 with original application based on text prediction. Federated Learning can be applied to GBMs though. A lot of new developments have arisen that we could look into next. It should be easy to apply GLMs in a Federated Learning context also, given that a GLM is basically a neural network with no hidden layers, and has widespread use in Life and Health insurance.

Another difficulty is data set heterogeneity. In our use case, each of the insurers had the exact same amount of data, but in reality that is not the case. One insurer might be a market leader with a significant market share. We are confident that Federated Learning can be applied even when the data is not uniformly split between ten insurers however, and there will be more work on that in future.

Dr Śmietanka: To summarise, we have shown that it is possible to collaborate on models without compromising privacy of data, and Federated Learning can be a technology that will help with that. Finally, we have made it work and shared the code on GitHub that makes systems collaborate on a Machine Learning model without sharing or exposing the raw data.

Moderator (starting the Q&A session): From your experience, where do you think this is likely to be used first within the actuarial profession on a sustainable basis, given the challenges you see for adoption?

Mr Liew: One of the more obvious opportunities would probably be the example we shared earlier on Life actuaries pooling mortality rates together in the CMI studies. It takes a while to anonymise the data, to securely transfer it over to the CMI, and for then for the CMI, as a trusted body, to do the aggregation, check, validate and publish it. Sometimes mortality and morbidity

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

rates have moved on by then. What we are trying to solve is calculating a shared average. I can imagine the IFoA and the CMI, as a trusted aggregator that most actuaries hopefully have a lot of faith in, could use this technology. Lloyd's syndicates often work with limited amounts of data in some lines of business and I can see perhaps Lloyd's of London as the marketplace, or the LMA, as a kind of trusted third party in the middle, orchestrating that. I have done a bit of operational risk modelling, and fraud is a problem shared by many insurers. It is in everyone's interest to have the best fraud model possible. My experience is that insurers, at least for fraud, are willing to collaborate with other insurers, and could build a fraud detection model together.

Mr Hand: With the CMI example, it typically takes seven or eight years for them to update the tables.

Dr Śmietanka: One non-obvious way of monetising data is sharing insights instead of raw data. Fraud is more natural because there is no competition. Everyone benefits from better fraud models.

Moderator: Apart from the examples you state, which other examples did you find particularly interesting?

Mr Liew: There are some Internet of Things devices. Samsung have fridges that are all trying to have failure rate detection models. They are trying to predict when they will go off and I think they have employed Federated Learning. You may have a fridge at home that is doing this!

Dr Śmietanka: And there are medical use cases, especially in the Covid period. One of the first examples, outside of mobile phones, was training Covid prediction models.

Mr Liew: This approach is flexible. It can work with neural networks, deep learning models and GBMs; but it equally works with deep LLMs, very large models, and Transformers. Given people's concerns around data privacy and what these LLMs are learning, Federated Learning will figure in some form too.

Moderator: Can Federated Learning accommodate heterogeneous models where each participant uses a slightly different architecture and how does this model performance get affected when each of the clients has different quality levels of data or different volumes of data?

Mr Liew: In terms of Horizontal Federated Learning that we presented here, one of the prerequisites is for every client, agent and insurer to use the same model form. It is not going to work if one person is using a GLM with 100 features and another person is using five features that are totally different as a GBM. There are variants of Transfer Learning, where you can train parts of the network and only share those parts. In terms of heterogeneity, the original 2016 Google paper, when they trained it on smartphones, has huge amounts of heterogeneous data. Different people text different amounts. Some people are serial texters and they use a different language. Sometimes, even with time zones, you can find that the data becomes quite different. In other domains, data can be updated frequently. You get a stream and flow second-by-second. When Google started to do Federated Learning on smartphones, they noticed that people might plug their phone in to initiate the training all in the UK time zone at some point. These text-prediction models would be trained on British English. As you went more towards America in terms of time zone, it would then use the American spelling and get that training data. Somehow, it held up in practice robustly against serial texters and different languages. In the text world, it has been robustly studied that it can deal with heterogeneity and holds up fairly well. To some extent, the Federated Learning aspect, the sharing and aggregation, is almost like a bit of regularisation. It avoids over-fitting, which can sometimes lead to federated models even performing better than a non-federated model. We have briefly tried in our working group to experiment with heterogeneous data splits and found the results to be fairly good. We would welcome anyone else to try it out for themselves.

Moderator: In the model fit you did and the comparison on the French third party liability dataset, what was your predictive variable there and what drove that choice?

Mr Liew: It was claims frequency. Ultimately, we were driven by the volume of existing literature. I would consider the third-party French data set a bit like the MNIST data set, which is

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

often used as a starting point or benchmark for actuarial science Machine Learning papers or techniques. A lot of people in the research field have a general view for what is a good error on that data. It makes it easy to compare and make sure we are fitting the model correctly. Also, the likes of Mario Wuthrich and Ronald Richman have studied this data quite a lot. We are very aware of the transformations and what sort of neural network architecture worked well. We did not want to spend too much time getting into the nitty gritty of hyperparameter tuning. That is not what this working party was about. We are not trying to eke out the best possible regression model. We are trying to prove that Federated Learning works. We thought that was the quickest possible route forward - to study a well-established data set with a good, reliable benchmark.

Moderator: As part of regularisation of data, would you also need to be agreeing on encodings across all the players and other feature changes and modifications?

Mr Liew: We did notice that was a challenge. If you follow the existing literature on this data set, it recommended doing transformations such as the Min–Max scaler, which works very well for a lot of activation functions in neural networks. In practice, that might mean all the insurers will have to somehow agree on the same minimum and maximum factors, like driver age, so that they can all agree to Min-Max scale using those parameters. In our paper, we did argue that, potentially, insurers won't want to share with fellow insurers what their youngest driver age is, or oldest. One thing you can do to help mitigate that limitation is use another privacy-enhancing technique called homomorphic encryption.

Mr Hand: It is a type of encryption technique where you can add, subtract, or multiply numbers, but the data remains encrypted.

Mr Liew: It is quite close to Federated Learning in the sense that you could take data, encrypt it and then gain insight from it, but you are only interrogating the encrypted data itself.

Moderator: Due to different underwriting methods or other issues, each of the players' modules or models has its own set of biases and specificities. One of the industry participants who has now got this Federated Model back will need to understand how can they reintegrate their customisations or their own specific context back into the model they will use in production?

Mr Liew: Use them as you would the Life tables when you get them in the CMI.

Mr Hand: We will adjust them based on our own experience and apply other techniques, such as GLMs. You can apply lots of different methods to get more granular assumptions.

Mr Liew: In life insurance, we take industry-averages, and tailor them depending on the perceived relative ranking of our own underwriting bias. As a working party, we think you could apply the same technique to a Federated Model. Take a claims frequency prediction or a mortality prediction from an industry average, then apply judgement or actual versus expected analysis retrospectively, to adjust that prediction to experience.

Dr Śmietanka: We want to stress that Federated Learning may not always work for everyone. It is a pipeline to train models collaboratively, but everyone should validate that model on their own data as well, particularly for specialised insurance.

Moderator: Is there a way that you could get to a point where one of the individual player models is better than the federated one?

Mr Liew: Potentially, depending on the size of the data set, the problem may be not worth doing. In markets where you do have a dominant player, you are getting watered down, to some extent, by a Federated Model and you probably would not want to participate. This is probably more of a use case where you have an even split of the market, where there are many competitive smaller insurers.

Moderator: You raised areas of potential further research between neural networks, embeddings, GBMs and gradient. Any idea which ones you would be most interested to pursue next?

Mr Liew: A lot of the feedback from our research paper has been around data heterogeneity. We have a lot of our code base set up to tinker with dividing the data into other splits, other than a simple split into 10 tranches of equal size. We can quite easily change that to say that one insurer

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press

has all the young drivers, and one has all the old drivers; or one has many high-risk high claimants and some only live in a particular region; and so on. We can then experiment on those bases.

Mr Hand: The data set heterogeneity should not be too difficult to deal with. We have already done some of it. GBMs will be quite a good one to look at, given that a lot of companies use this already.

Mr Liew: What I suspect might be a barrier to adoption is regulation. The regulation has not caught up to what is and isn't allowed on Federated Learning. Most privacy experts are quite clear that you cannot share names and addresses that are PII. However, on the question of whether encrypted model parameters using secure multi-party computation can be shared, I am not sure the answer is written down in GDPR. I do not know if it has yet been covered explicitly or if there is some other data regulation.

Moderator: Thank you all very much.

Cite this article: Hush hush: Keeping neural networks claims modelling private, secret, and distributed using federated learning: Discussion. *British Actuarial Journal*. https://doi.org/10.1017/S1357321726100415

https://doi.org/10.1017/S1357321726100415 Published online by Cambridge University Press