Field-regularised Factorization Machines for Mining the Maintenance Logs of Equipment

Zhibin Li[1], Jian Zhang[1], Qiang Wu[1], and Christina Kirsch[2]

[1] University of Technology Sydney, Sydney, Australia
Zhibin.Li@student.uts.edu.au, Jian.Zhang@uts.edu.au, Qiang.Wu@uts.edu.au
[2] Sydney Trains-Operational Technology, Sydney, Australia
Christina.Kirsch@transport.nsw.gov.au

Abstract. Failure prediction is very important for railway infrastructure. Traditionally, data from various sensors are collected for this task. Value of maintenance logs is often neglected. Maintenance records of equipment usually indicate equipment status. They could be valuable for prediction of equipment faults. In this paper, we propose Field-regularised Factorization Machines (FrFMs) to predict failures of railway points with maintenance logs. Factorization Machine (FM) and its variants are state-of-the-art algorithms designed for sparse data. They are widely used in click-through rate prediction and recommendation systems. Categorical variables are converted to binary features through one-hot encoding and then fed into these models. However, field information is ignored in this process. We propose Field-regularised Factorization Machines to incorporate such valuable information. Experiments on data set from railway maintenance logs and another public data set show the effectiveness of our methods.

Keywords: Factorization Machines · Failure prediction · Categorical data.

1 Introduction

Railway points are a kind of mechanical installations allowing railway trains to be guided from one track to another. They are among the key components of railway infrastructure. As a part of the signal equipment, points control the routes of trains at railway junctions, having a great impact on the reliability and punctuality of rail transport. Existing research on failure prediction of points mainly relies on additional sensors' data [1, 6, 7, 15, 22, 26], e.g. voltages, currents and forces. Installation of sensors incurs costly labour and material expenses, as well as the possibility of sensor malfunction, which limits their implementation. Other research focuses on approximating the long-term degradation curve of equipment under certain maintenance strategy [11, 12, 18, 21, 23], rather than predicting failure of equipment in the near future.

Maintenance logs of equipment contain formatted maintenance records, including maintenance type, components, finished time, etc. They can be of great

value in failure prediction. These data often carry information of equipment status with timestamps. Compared to data collected by sensors, maintenance records are usually ready to hand with a specified format. They mainly consist of categorical variables and could be very sparse after commonly performed one-hot encoding. Besides, railway points consist of many components, and failures can be viewed as a result of their interactions. Domain knowledge regarding such interactions might be very limited and depends on equipment types. In order to predict failures with maintenance logs, the model needs to learn the complex interactions from such sparse data.

Aiming at this challenging task, we put forward Field-regularised Factorization Machines (FrFMs) for failure prediction of railway points. Factorization Machines (FMs) combine the advantages of Support Vector Machines (SVMs) with factorization models [19]. In contrast to SVM, FMs factorise all interactions between features into products of two low-rank matrices. In this way, they are likely to learn interactions which even do not appear in training data. Many variants of FMs have been proposed and achieved good performance. Locally Linear Factorization Machines [13] adopts locally linear coding scheme and jointly optimise FM models with anchor points. They are capable of learning complex non-linear data by exploring local coding technique. Wang et al. [24] propose Contextual and Position-Aware Factorization Machines targeted at sentiment analysis of texts. Inspired by the neural skip-gram model, Contextual and Position-Aware Factorization Machines limits interactions to a range of words. In addition, latent vectors are learned based on the relative position of words, which means that there will be several independent latent vectors for one word. FMs are usually limited to quadratic models, and related loss functions are non-convex. Many papers have focused on overcoming these two limitations. Neural Factorization Machines [9] take in the advantages of deep neural networks to modelling higher-order feature interactions. They firstly encode feature vectors by pre-training FMs and then train a neural network with these embedding vectors. DeepFM [8] is similar to Neural Factorization Machines, except that it is an end-to-end model that requires no pre-training. Unlike Neural Factorization Machines, DeepFM jointly learns the embedded vectors and the neural networks. Yamada et al. [25] reformulate the optimisation problem of FMs as a semi-definite programming problem. By introducing nuclear norm in FMs, their loss functions of FMs becomes convex.

The above-mentioned models focus less on the inherent properties of data carried by field information. Field-aware Factorization Machines (FFMs) [10] consider the field structure of data and learn pair-wise interactions with regard to each pair of fields. They are more complex than FMs in terms of the number of parameters and computational complexity. Field-weighted Factorization Machines [16] add additional coefficients to depict the interactions of fields, and reduce the number of model parameters compared to FFMs. These models treat features from different fields differently. In other words, they only consider inter-field information.

Existing models either ignore the field information or only consider the inter-field information. They neglect the **relationships among features inside each field**, which is going to be used in our models.

Our contributions could be shown in two aspects. Firstly, to the best of our knowledge, it is the first time that maintenance logs are used to predict the failure of railway points. Secondly, we propose FrFMs which leverage field information and develop a method to solve the related optimisation problems. Experiments on two data sets show that our methods can achieve better performance compared to some state-of-the-art methods.

2 Preliminaries

A degree-2 polynomial mapping can often effectively capture the information of feature conjunctions [2]. It learns a weight for each feature conjunction:

$$\phi_{Poly2}(W, \boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{i,j} x_i x_j$$

$$W = (w_{i,j}) \in \mathbb{R}^{n \times n}, \boldsymbol{x} \in \mathbb{R}^n \tag{1}$$

where W is the learned weight matrix and $\boldsymbol{x}$ is the input vector of dimension n. Corresponding 2-way FMs can be written in following form:

$$\phi_{FM}(V, \boldsymbol{x}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \boldsymbol{v}_i, \boldsymbol{v}_j \rangle x_i x_j$$

$$V = \begin{bmatrix} \boldsymbol{v}_1 \\ \boldsymbol{v}_2 \\ \vdots \\ \boldsymbol{v}_n \end{bmatrix} \in \mathbb{R}^{n \times k}, \boldsymbol{x} \in \mathbb{R}^n \tag{2}$$

$\langle \cdot, \cdot \rangle$ stands for dot product of two vectors. $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ denote two row vectors of V with dimension k. $\boldsymbol{v}_i$ is referred to as **embedding vector** or **latent vector** for feature i. For simplicity of formulations, we omit linear terms and bias term following [10], but we include them in experiments.

Categorical data are highly sparse after one-hot encoding. Some pairs of $x_i x_j$ might even not appear in training data. In this case, for polynomial mapping some $w_{i,j}$ are not able to be learned. By factorizing weight matrix W into VV^T, FMs are able to learn interactions for rare feature pairs. Each row vector $\boldsymbol{v}_i$ in V stands for latent vector regarding feature x_i.

3 Field-regularised Factorization Machines

3.1 Motivation

Table 1 presents some simple data constructed from maintenance records for failure prediction. 'Maintenance Type' and 'Component' are two different **fields**.

Table 1. A sample of maintenance records with failures to be predicted.

Failure	Maintenance Type	Component
1	A	II
1	C	II
-1	B	VI

A, B and C stand for different maintenance types that can probably be 'Routine Inspection', 'Corrective Maintenance' and so on. The field 'Component' shows the maintenance was performed over which component. '1' and '-1' in column 'Failure' stand for whether there was a fault occurred after this maintenance and before next planned maintenance.

FMs will learn latent vectors for A, B, C, II and VI respectively. In engineering practice, we anticipate different effects with different maintenance behaviours. Each field can be regarded as a classification criterion for maintenance work, and corresponding features in that field are the class labels. We would prefer diverse latent vectors in the same field because we could distinguish the effects caused by different maintenance work in this way. As a result, latent vectors for A, B and C should be diverse, as well as latent vectors for II and VI.

3.2 Methods

In this section, we propose the FrFMs for binary classification. For simplicity of formulations, we omit linear terms and bias term following [10], but we include them in experiments as they often improve the results. The loss function of FrFMs with logistic loss regarding one sample $(y, \boldsymbol{x})$ is:

$$\mathcal{L}(V) = \log(1 + \exp(-y\phi_{FM}(V, \boldsymbol{x}))) + \frac{\lambda_1}{2}\|V\|_F^2 + \frac{\lambda_2}{2}R(V) \tag{3}$$

$\phi_{FM}(V, \boldsymbol{x})$ is defined in (2), as we share the same prediction function with FMs. $\|\cdot\|_F$ is the Frobenius norm for matrices. $y \in \{-1, 1\}$ is the ground truth label for sample $\boldsymbol{x}$. The first term denotes the prediction loss compared to ground truth, and the second term forces the solution V sparse. $R(V)$ is a regulariser that measures the similarity of latent vectors in each field, and we prefer smaller similarity as discussed above. By introducing $R(V)$ into loss function, field information is included. λ_1, λ_2 are two non-negative parameters obtained by cross validation.

In order to capture the inherent properties come with fields of data, we construct a feature relation matrix A which will be included in $R(V)$:

$$A_{i,j} = \begin{cases} \frac{1}{N_{i,j}} & \text{if } x_i, x_j \text{ are in same field and } i \neq j, \\ 0 & \text{else.} \end{cases} \tag{4}$$

$N_{i,j}$ is the number of features in the field contains x_i and x_j. It is introduced to avoid deviation caused by different number of features in different fields. Each

element in A stands for the relationship of two features. If they are in same field, then corresponding entries in A will be one divided by the number of features in this field. Otherwise they will be zeros.

Various metrics can be used to measure the similarity of latent vectors. In this work, we will present FrFM with Euclidean distance and cosine similarity.

FrFM-EUC We refer to FrFM with Euclidean distance as FrFM-EUC. Euclidean distance is used to measure the similarity of two vectors in FrFM-EUC, and larger Euclidean distance indicates smaller similarity. Therefore, $R(V)$ has the following form:

$$R(V) = -\sum_{i=1}^{n}\sum_{j=i+1}^{n} A_{i,j}\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 \tag{5}$$

$\|\cdot\|_2$ denotes l^2-norm for vectors. The loss function for FrFM-EUC is:

$$\mathcal{L}_{euc}(V) = \log(1+\exp(-y\phi_{FM}(V,\boldsymbol{x}))) + \frac{\lambda_1}{2}\|V\|_F^2 - \frac{\lambda_2}{2}\sum_{i=1}^{n}\sum_{j=i+1}^{n} A_{i,j}\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 \tag{6}$$

FrFM-COS FrFM-COS denotes FrFM with cosine similarity. $R(V)$ has the following form:

$$R(V) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}\frac{\langle \boldsymbol{v}_i, \boldsymbol{v}_j\rangle}{\|\boldsymbol{v}_i\|_2\|\boldsymbol{v}_j\|_2} \tag{7}$$

Directly optimizing (3) with (7) is complicated. Rewriting rows of V into products of their direction vectors and lengths leads to:

$$V = \begin{bmatrix} w_1\hat{\boldsymbol{v}}_1 \\ w_2\hat{\boldsymbol{v}}_2 \\ \vdots \\ w_n\hat{\boldsymbol{v}}_n \end{bmatrix} \in \mathbb{R}^{n\times k}, \quad \hat{\boldsymbol{v}}_i = \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|_2}, \quad w_i = \|\boldsymbol{v}_i\|_2 \tag{8}$$

Then (7) equals to:

$$R(V) = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{i,j}\hat{\boldsymbol{v}}_i\hat{\boldsymbol{v}}_j^T = tr(\hat{V}^T A\hat{V}) \tag{9}$$

Substitute V with $\hat{V}$ and $\boldsymbol{w}$ in formulation of FMs:

$$\phi_{FM}(\hat{V},\boldsymbol{w},\boldsymbol{x}) = \sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle w_i\hat{\boldsymbol{v}}_i, w_j\hat{\boldsymbol{v}}_j\rangle x_i x_j \tag{10}$$

and finally we get loss function for FrFM-COS:

$$\begin{aligned}\mathcal{L}_{cos}(\hat{V},\boldsymbol{w}) = \log(1+\exp(-y\phi_{FM}(\hat{V},\boldsymbol{w},\boldsymbol{x}))) + \frac{\lambda_1}{2}\|\boldsymbol{w}\|_2^2 + \frac{\lambda_2}{2}tr(\hat{V}^T A\hat{V}) \\ s.t.\ \|\hat{\boldsymbol{v}}_i\|_2 = 1,\ \forall i = 1,2,...,n. \quad \boldsymbol{w}\in\mathbb{R}_+^{1\times n}\end{aligned} \tag{11}$$

3.3 Optimization

Similar to FMs, our loss functions are non-convex. Gradient descent is used to find local minima of our loss functions. Stochastic Gradient Descent (SGD) is widely used in optimisation of FMs and its variants. It has shown its effectiveness. Mini-batch Gradient Descent also enjoys the advantages of SGD while it is more efficient. Thus we adopt Mini-batch Gradient Descent in optimisation. We apply AdaGrad [5] to determine the learning rate in each iteration for it has shown great power in similar problems [3,10]. To lessen over-fitting, we utilise early-stop strategy in training of FrFM-EUC and FrFM-COS. The best training epoch T will be decided based on a validation set.

FrFM-EUC The gradient with regard to one sample $(y, \boldsymbol{x})$ is:

$$\begin{aligned}\frac{\partial \mathcal{L}_{euc}(V)}{\partial \boldsymbol{v}_i} =& \frac{-y}{1+\exp(y\phi_{FM}(V,\boldsymbol{x}))}(x_i \sum_{j=1}^{n} \boldsymbol{v}_j x_j - \boldsymbol{v}_i x_i^2) \\ &+ (x_i \neq 0)(\lambda_1 \boldsymbol{v}_i - \lambda_2 \sum_{j=1}^{n} A_{i,j}(\boldsymbol{v}_i - \boldsymbol{v}_j))\end{aligned} \tag{12}$$

$(x_i \neq 0)$ in (12) indicates that gradients would be zero if corresponding features are zero. This strategy has been used in FFMs and performs well. We can update model parameters with adaptive learning rate in iteration l:

$$G_{i,f}^{(l+1)} = G_{i,f}^{(l)} + (\frac{\partial \mathcal{L}_{euc}(V)}{\partial v_{i,f}}\Big|_{V=V^{(l)}})^2 \tag{13}$$

$$v_{i,f}^{(l+1)} = v_{i,f}^{(l)} - \frac{\eta}{\sqrt{G_{i,f}^{(l+1)} + \epsilon}} \circ \frac{\partial \mathcal{L}_{euc}(V)}{\partial v_{i,f}}\Big|_{V=V^{(l)}} \tag{14}$$

$\circ$ denotes element-wise multiplication of vectors. G stores the accumulated square gradient for AdaGrad and ϵ is s a smoothing term that avoids division by zero (we set it to 10^{-8} in this paper). The training process for FrFM-EUC is presented in Algorithm 1.

Algorithm 1 Training FrFM-EUC by Mini-batch Gradient Descent

input Data matrix $D \in \mathbb{R}^{M \times n}$ contains M samples, feature relation matrix A, latent dimension k, hyper-parameters λ_1, λ_2, learning rate η, batch size m, $G^{(0)} = \mathbf{0}$.
Randomly initialise $V^{(0)} \in \mathbb{R}^{n \times k}$ with values sampled from a uniform distribution $[0, 1/\sqrt{k}]$. Calculate the number of batches $b = \lfloor \frac{M}{m} \rfloor$.
for *Epoch* $= 0$ to T **do**
  Shuffle the samples in D randomly.
  Split D into batches $X_1, X_2, ..., X_b \in \mathbb{R}^{m \times n}$.
  for $i \in \{1, 2, ..., b\}$ **do**
    Calculate the gradient of V by (12) for every sample in X_i and get the average.
    Update accumulated square gradient G by (13).
    Update V by (14).

FrFM-COS The gradient with regard to one sample $(y, \boldsymbol{x})$ is:

$$\begin{aligned}\frac{\partial \mathcal{L}_{cos}(\hat{V}, \boldsymbol{w})}{\partial \hat{\boldsymbol{v}}_i} =& \frac{-y}{1+\exp(y\phi_{FM}(\hat{V}, \boldsymbol{w}, \boldsymbol{x}))}(w_i x_i \sum_{j=1}^{n} \hat{\boldsymbol{v}}_j w_j x_j - \hat{\boldsymbol{v}}_i w_i^2 x_i^2) \\ &+ (x_i \neq 0)\lambda_2 \sum_{j=1}^{n} A_{i,j} \hat{\boldsymbol{v}}_j\end{aligned} \tag{15}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_{cos}(\hat{V}, \boldsymbol{w})}{\partial \boldsymbol{w}} =& \frac{-y}{1+\exp(y\phi_{FM}(\hat{V}, \boldsymbol{w}, \boldsymbol{x}))}((\boldsymbol{w} \circ \boldsymbol{x})(\hat{V}\hat{V}^T - diag(\hat{V}\hat{V}^T))) \circ \boldsymbol{x} \\ &+ \lambda_1(\boldsymbol{x} \neq 0) \circ \boldsymbol{w}\end{aligned} \tag{16}$$

$(\boldsymbol{x} \neq 0)$ is a binary row vector indicates non-zero indices of $\boldsymbol{x}$. Similarly, gradients would be zero if corresponding features are zero. With gradient in hand, we can train the model similar to Algorithm 1. Differences are that we need to project $\hat{V}$ and $\boldsymbol{w}$ into feasible sets in each iteration.

4 Experiments

4.1 Data Set

POINTS-3 data set was generated from the maintenance logs of Sydney Trains' railway points. For numerical features, they were simply transformed into features 'Zero' or 'Non-Zero'. As shown in Fig. 1, for one piece of equipment, we selected three consecutive maintenance records: Maintenance 1, Maintenance 2 and Maintenance 3, to construct a sample and labelled the sample depending on whether a failure occurred between Maintenance 3 and Maintenance 4. If there was a failure record, then this sample was labelled with '1', otherwise '-1'.

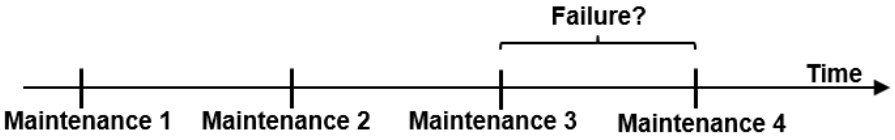

Fig. 1. An example for sample labelling in POINTS-3.

Equipment details including equipment type, location and other features were also concatenated to construct one data sample. We randomly split the data set into 60% training set, 20% validation set and 20% test set.

Phishing data set contains important features that have been proven to be sound and effective in predicting phishing websites [4]. We randomly split the data into 64% training set, 16% validation set and 20% test set.

Table 2 summarises the statistics of the data sets.

Table 2. Statistics of the data sets.

Data Set	# Instances	# Features	# Fields
POINTS-3	55784	2226	52
Phishing	11055	68	30

4.2 Baselines and Hyper-parameter Tuning

We compare our models with three baselines.

LINEAR-LR denotes Logistic Regression with linear terms. It has been proven to be effective in classification tasks with sparse data. We implemented LINEAR-LR with Python library sklearn [17].

FM is the implementation of Factorization Machines defined in (2). We also included linear terms and bias term.

FFM is the implementation of Field-aware Factorization Machines. We also included linear terms and bias term.

FrFM-EUC and **FrFM-COS** stand for our methods proposed in this paper.

Both FM and FFM were implemented by xLearn [14] with AdaGrad and SGD optimizer. All hyper-parameters were chosen based on validation sets. The regularisation parameters were chosen from $\{10^{-6}, 10^{-5}, ..., 10^{6}\}$ for LINEAR-LR and $\{10^{-6}, 10^{-5}, ..., 10^{-1}\}$ for all other methods. Learning rates for AdaGrad were chosen from $\{0.02, 0.2\}$. Latent dimensions were chosen from $\{20, 40, ..., 100\}$ for FM and our method, and from $\{10, 20, ..., 50\}$ for FFM. Early-stop strategy was adopted for FM, FFM and our method to reduce over-fitting. Batch size was set to 64 in training of FrFM-EUC and FrFM-COS.

4.3 Results and Metrics

Metrics We calculated Logloss of each baseline on every data set. **Logloss** is given by:

$$\text{Logloss} = \frac{1}{M}\sum_{i=1}^{M} log(1 + \exp(-y_i\hat{y}_i)) \tag{17}$$

y_i and $\hat{y}_i$ are the label and model output for test sample i respectively. M is the total number of test instances.

AUROC and **AUPRC** stand for area under receiver operating characteristic curve and area under precision-recall curve respectively.

Results Table 3 shows the results on different data sets, the best results are bold and second best are underlined. We trained and tested these models five times on each data set and reported the average results. POINTS-3 data set is an imbalanced data set with only 1701 positive samples out of 55784 samples, so AUPRC is more representative compared to AUROC according to [20]. AUPRC were calculated from $recall > 0.1$ for the reason that too low recall is meaningless in our case. Phishing data set is a balanced data set that won't show much difference between AUROC and AUPRC, so we only present the AUROC for it.

Table 3. Comparison of LINEAR-LR, FM, FFM, FrFM-EUC and FrFM-COS.

Method	POINTS-3			Phishing	
	AUROC	AUPRC ($recall > 0.1$)	Logloss	AUROC	Logloss
LINEAR-LR	0.7012	0.0641	0.1275	0.9886	0.1384
FM	0.6987	0.0622	0.1285	0.9911	0.1226
FFM	0.6974	0.0619	0.1291	0.9923	0.1134
FrFM-COS	0.7090	**0.0676**	0.1271	0.9925	0.1120
FrFM-EUC	**0.7108**	0.0674	**0.1270**	**0.9950**	**0.0919**

Experiment results show that our methods perform best on these two data sets. Precision-recall curves related to POINTS-3 data set for $recall > 0.1$ and $precision > 0.06$ are plotted in Fig. 2.

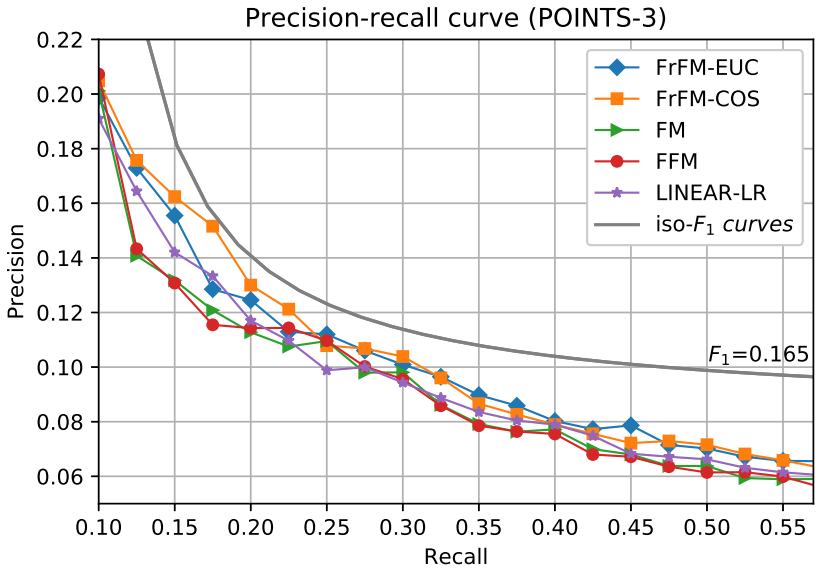

Fig. 2. Precision-recall curves with regard to POINTS-3.

Fig. 2 shows that FrFM-COS can also achieve the best F_1-score (0.165) compared to other methods. By appropriately setting threshold value for the classifier got from FrFM-EUC, we can get an overall **Accuracy: 90.99%**, with **Precision: 11.02%** and **Recall: 27.65%**. This may not be a perfect prediction but it is still acceptable considering that we didn't use any sensor data (e.g. current,

voltage, force and so on). There are wrongly recorded data and failures that are caused by vandalism which makes some failures unpredictable. Outputs of the model could be used as references for maintenance plans.

Receiver operating characteristic curves with regard to Phishing data set are plotted in Fig. 3. Our method FrFM-EUC consistently outperforms other methods.

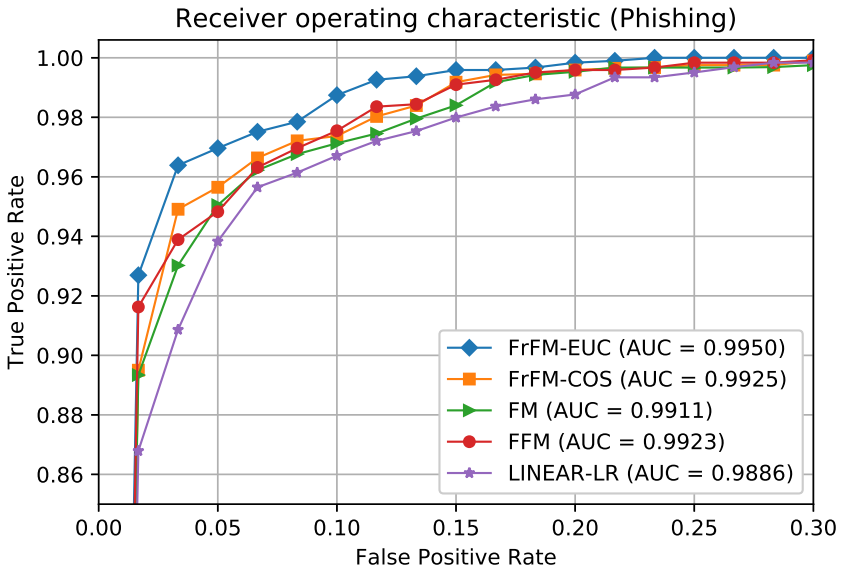

Fig. 3. Receiver operating characteristic curves with regard to Phishing.

5 Conclusion and Future Work

In this paper, we proposed the Field-regularised Factorization Machines for failure prediction of railway points. Field information is often ignored in many related methods. Especially for the inner-field relationships among features, there is little work concerning them. The key components of FrFMs are the regularisation terms that incorporate field information in the training process. Two forms of FrFMs: FrFM-EUC and FrFM-COS are presented. Experiment results showed that our models outperformed some state-of-the-art methods in predicting failure of railway points. We also achieved a better result on a public data set.

The predictions for points failure were not perfect but could be used as the reference for maintenance plans. Our following work will be focusing on combin-

ing data from other sources with maintenance data to improve the prediction results.

Acknowledgements. The authors greatly appreciate the financial support from the Rail Manufacturing Cooperative Research Centre (funded jointly by participating rail organisations and the Australian Federal Governments Business Cooperative Research Centres Program) through Project R3.7.2 - Big data analytics for condition based monitoring and maintenance.

References

1. Camci, F., Eker, O.F., Başkan, S., Konur, S.: Comparison of sensors and methodologies for effective prognostics on railway turnout systems. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit **230**(1), 24–42 (2016)
2. Chang, Y.W., Hsieh, C.J., Chang, K.W., Ringgaard, M., Lin, C.J.: Training and testing low-degree polynomial data mappings via linear svm. Journal of Machine Learning Research **11**(Apr), 1471–1490 (2010)
3. Chin, W.S., Zhuang, Y., Juan, Y.C., Lin, C.J.: A learning-rate schedule for stochastic gradient methods to matrix factorization. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 442–455. Springer (2015)
4. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
5. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research **12**(Jul), 2121–2159 (2011)
6. García Márquez, F.P., Roberts, C., Tobias, A.M.: Railway point mechanisms: condition monitoring and fault detection. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit **224**(1), 35–44 (2010)
7. Guclu, A., Yilboga, H., Eker, Ö.F., Camci, F., Jennions, I.K.: Prognostics with autoregressive moving average for railway turnouts (2010)
8. Guo, H., Tang, R., Ye, Y., Li, Z., He, X.: Deepfm: a factorization-machine based neural network for ctr prediction. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 1725–1731. AAAI Press (2017)
9. He, X., Chua, T.S.: Neural factorization machines for sparse predictive analytics. In: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 355–364. ACM (2017)
10. Juan, Y., Zhuang, Y., Chin, W.S., Lin, C.J.: Field-aware factorization machines for ctr prediction. In: Proceedings of the 10th ACM Conference on Recommender Systems. pp. 43–50. ACM (2016)
11. Kobayashi, K., Kaito, K., Lethanh, N.: A bayesian estimation method to improve deterioration prediction for infrastructure system with markov chain model. International Journal of Architecture, Engineering and Construction **1**(1), 1–13 (2012)
12. Le Son, K., Fouladirad, M., Barros, A.: Remaining useful lifetime estimation and noisy gamma deterioration process. Reliability Engineering & System Safety **149**, 76–87 (2016)
13. Liu, C., Zhang, T., Zhao, P., Zhou, J., Sun, J.: Locally linear factorization machines. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence. pp. 2294–2300. AAAI Press (2017)

14. Ma, C.: xlearn, https://github.com/aksnzhy/xlearn
15. Oyebande, B., Renfrew, A.: Condition monitoring of railway electric point machines. Iee Proceedings-Electric Power Applications **149**(6), 465–473 (2002)
16. Pan, J., Xu, J., Ruiz, A.L., Zhao, W., Pan, S., Sun, Y., Lu, Q.: Field-weighted factorization machines for click-through rate prediction in display advertising. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web. pp. 1349–1357. International World Wide Web Conferences Steering Committee (2018)
17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
18. Rama, D., Andrews, J.D.: A reliability analysis of railway switches. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of rail and rapid transit **227**(4), 344–363 (2013)
19. Rendle, S.: Factorization machines. In: Data Mining (ICDM), 2010 IEEE 10th International Conference on. pp. 995–1000. IEEE (2010)
20. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. PloS one **10**(3), e0118432 (2015)
21. Shafiee, M., Patriksson, M., Chukova, S.: An optimal age–usage maintenance strategy containing a failure penalty for application to railway tracks. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit **230**(2), 407–417 (2016)
22. Tao, H., Zhao, Y.: Intelligent fault prediction of railway switch based on improved least squares support vector machine. Metallurgical and Mining Industry **7**(10), 69–75 (2015)
23. TSUDA, Y., KAITO, K., AOKI, K., KOBAYASHI, K.: Estimating markovian transition probabilities for bridge deterioration forecasting. Structural Engineering/Earthquake Engineering **23**(2), 241s–256s (2006)
24. Wang, S., Zhou, M., Fei, G., Chang, Y., Liu, B.: Contextual and position-aware factorization machines for sentiment classification. arXiv preprint arXiv:1801.06172 (2018)
25. Yamada, M., Lian, W., Goyal, A., Chen, J., Wimalawarne, K., Khan, S.A., Kaski, S., Mamitsuka, H., Chang, Y.: Convex factorization machine for toxicogenomics prediction. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1215–1224. ACM (2017)
26. Yilboga, H., Eker, Ö.F., Güçlü, A., Camci, F.: Failure prediction on railway turnouts using time delay neural networks. In: 2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA). pp. 134–137. IEEE (2010)