A CNN-based Hybrid Model and Architecture for Shilling Attack Detection

Mahsa Ebrahimian
Electrical, Computer, and Biomedical Engineering Department
Ryerson University Toronto, Canada
mahsa.ebrahimian@ryerson.ca

Rasha Kashef
Electrical, Computer, and Biomedical Engineering Department
Ryerson University, Toronto, Canada
rkashef@ryerson.ca

2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) | 978-1-6654-4864-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/CCECE53047.2021.9569048

***Abstract*—Recommendation systems are widely used in various areas to personalize recommendations and suggestions to users. However, they are vulnerable to shilling attacks in which malicious users try to promote their products or diminish their competitors'. Therefore, detecting shilling attacks can significantly improve the quality of recommender systems and user experience. With the increasing complexity of attacks and changes in attackers' behavior, more advanced approaches are required to find the hidden patterns in data. This paper proposes a CNN-based hybrid model and architecture to integrate self-learning and flexible aspects of CNN with other approaches to enhance the prediction results of shilling attacks. Two benchmark datasets are used in the experimental analysis, the Movie-Lens 100K and Netflix. The performance of the proposed hybrid models is compared to that of the traditional deep learning and machine learning detection methods. Experimental results show that the superiority of hybrid models over individual models depends on the sparsity level of data and divergence of results.**

Keywords—Recommendation systems, Shilling attacks, Deep learning, Neural Networks, Compatibility, Divergence Criterion.

I. INTRODUCTION

Nowadays, Recommendation systems (RS) are critical in the online journey of users who are overwhelmed by the information overload problem. RS facilitate users' decision-making by suggesting relevant items such as movie or music recommendation [1]. An efficient RS could be beneficial not only for users but also for companies to stand out significantly from competitors and generate a considerable amount of revenue. Collaborative filtering, a special kind of RS, assumes that people who agreed in the past will agree in the future and recommends personalized items using the similarities in their preferences. Thus, these types of RS are more vulnerable to shilling attacks. Shilling attacks occur when profit-driven users try to manipulate the results of a recommender in favor of their items by injecting fake ratings into the system. The intention is to elevate their targeted items or undermine their competitors'. Shilling attacks could adversely affect user's experience and result in revenue loss for the company. Therefore, shilling attack detection is essential to minimize the effect on the targeted items and offer more precise recommendations. The existing traditional detection methods such as statistical methods [2]graph mining [3], clustering [4], and classification [5] cannot be generalized on all variations of attack types [6], models, and attack sizes [7][8][9]. Although machine learning methods have shown significant advancements in shilling attack detection, their performance is dependent on a complicated and time-consuming feature extraction process to distinguish attacks and normal users based on the user profiles. Therefore, there is a need for a more robust and end-to-end solution to detect complex and divergent attacks in real-time. Deep learning methods are alternatives to efficiently detect the hidden patterns in a large number of ratings by undertaking the attributes derived from user profiles. Currently proposed deep learning methods on shilling attack detection include Convolutional Neural Network (CNN) [10][11], Recurrent Neural Network (RNN) [11][12][13] and Gated Recurrent Units (GRU) [14]. These methods have shown improvements in detecting obfuscated attack types and higher accuracy compared to traditional machine learning methods. Their simple network architecture still confines them in accounting for both the temporal and spatial information in the RS ratings. Therefore, combining the high-performing deep learning models could overcome the mentioned limitations. In this paper, an architecture is proposed to combine CNN with four other deep learning models, including Long Short-Term Memory (LSTM) Recurrent Neural Network, the Gated Recurrent Units (GRU), Extreme Gradient Boosting (XGBoost), and Spiking Neural Network (SNN) along with a comprehensive comparison between their performances on two benchmark datasets Movie-Lens 100K[1] and Netflix[2]. The main contributions of this paper are summarized as:

(1) We introduce a novel architecture to combine CNN and four other models to better detect shilling attacks in user-based collaborative filtering RS.
(2) We provide a comprehensive performance comparison between our CNN- based hybrid models.
(3) We introduce a new metric, F-divergence, to measure the compatibility of the hybrid to enhance the accuracy of the shilling attack detection result.

The rest of the paper is organized as follows: In Section 2, related work and background on recommendation systems and the shilling attack problem are introduced. In Section 3, our adopted models are discussed. We have introduced our proposed CNN- based hybrid model in Section 4. In section 5, The experimental results are provided. Finally, Section 5 concludes the paper and provides future directions.

II. RELATED WORK

A. Deep Learning Methods on Shilling Attacks

Deep learning methods learn directly from rating data and discover the hidden patterns in more complicated attack types. Recent studies have used deep learning methods as a supervised shilling detection model. Different CNN models are proposed for this purpose. For instance, authors in [10] tested a CNN

[1] https://grouplens.org/datasets/movielens/100k/

[2] https://www.kaggle.com/netflix-inc/netflix-prize-data/data

model on a small benchmark dataset. At the same time, the work in [11][12] have developed a CNN model using the largest version of the benchmark dataset and achieved high performance on Average Over Popular (AOP) attack types. But their training time is not computationally scalable [10]. Some have used RNN methods and applied time interval analysis on user rating data. A particular RNN model known as LSTM is proposed in [13] to predict items' ratings based on historical data. Comparing the predicted rate with the actual ones indicates if an item is targeted by shilling attacks. But they have not considered the number of misclassified items. The proposed LSTM model in [11] considers users' rating as sequential data and classifies the items into under attack and normal. Two types of RNN models, LSTM and GRU, are compared in [14] with similar structures using 32 hidden neurons and Softmax activation function to classify items, and LSTM returned higher F1-Measures. Unsupervised learning methods such as Boosted decision tree are also used [15] and showed high accuracy of up to 99%. Unsupervised deep learning methods are combined with community detection methods to detect attack profiles [16]. Based on the results obtained from the above studies, the time distribution of ratings in RS can reveal unusual patterns and the correlation between users' rating data. Therefore, single CNN and RNN models cannot consider all these aspects in detecting attacks, while hybrid models might overcome these limitations. Deep learning methods are used to create more complicated attack types. A graph convolutional neural network is proposed in [17] to balance the attacks' feasibility and effectiveness.

B. Hybrid Deep Learning Methods

In this section, the state-of-the-art hybrid deep learning models on shilling attack detection are presented with their merits and impediments. Reference [18] proposed a hybrid model to identify shilling attacks in the social-aware networks, combining two CNN and LSTM models. The hybrid model consists of single convolution, single transformation, pooling, and output layers followed by LSTM layers. The model has shown superior accuracy and execution time in comparison with benchmark hybrid models. But there is no comparison between single and proposed hybrid models to justify the extra computational cost of combining the models. The hybrid CNN-LSTM model proposed in [14] has a relatively lower number of LSTM layers and has shown better performance than the proposed CNN-GRU model. It is constructed based on a flexible time segmentation using a 3D array of users, items, and days.

III. ADOPTED LEARNING ALGORITHMS

This section discusses several learning models, including LSTM, GRU, SNN, and a decision tree-based ensemble machine learning (XGBoost). All the models are considered as binary classifications for genuine and attackers.

GRU: RNN models are shown to perform well on shilling attack detection problem [11] and inspired by [20] which showed the GRU model's superior performance over LSTM, we have applied it on shilling attack detection. In this architecture, considering ratings in time sequence, the GRU cell forgets a portion of trained rating information and updates it using a reset gate. The GRU layer is used similar to the LSTM layer with 100 hidden neurons and a Softmax activation function to classify users into normal and attackers.

XGBoost: is an expansible machine learning model for tree boosting. This flexible technique is used for classification and regression problems [21]. It uses an ensemble of classification and regression trees, each having K nodes. The objective of our XGBoost model is a binary logistic with 100 estimators, and the maximum depth of the tree is set to 4. To prevent overfitting, a step size shrinkage of 1 is used after each boosting step.

LSTM: User's ratings can be examined as sequential data, considering ratings of user i on item j at time t. Attackers are willing to schedule it in the short term to balance the cost and effect of the attacks. Therefore, RNN can be used to train data considering the time intervals analysis. When the time sequence extends, the memory ability of the later-time node will decrease. However, the LSTM answers this problem by introducing a cell element. The proposed LSTM model has similar many to one process sequence in the [19] sentiment classification. There are two layers in this network, consisting of 100 hidden neurons in the LSTM layer followed by a Softmax activation function in the dense layer.

SNN is the third generation of neural networks that mimics the actual mechanisms of our brain's neurons and has a new type of neural model called spiking neuron. Spikes are discrete events during a specific time. The temporal dynamic of SNNs makes it computationally plausible as real-time solutions. Unlike the previous generation of neural networks, it does not update weights [22]. SNN has never been used for detecting shilling attacks to the best of our knowledge. Therefore, SNN could be a good alternative for shilling attack detection considering the temporal information in RS and the need to detect attacks in real-time. The SNN model adopted in this paper is inspired by [23]. We have used Poisson encoding for spikes generation based on the rating matrix. The rating matrix is normalized between 0 and 1 before feeding into the encoding process. The learning process is done by comparing desired and output spike trains using spike-timing-dependent plasticity (STDP). The number of Poisson input neurons is set to 100 in steps of 50, and each simulation ran 1,000 milliseconds. In addition, to improve the SNN model's performance on a sparse dataset [24] such as the RS rating matrix, we have used SMOTE method with majority class under-sampling of instances prior to learning.

IV. THE PROPOSED CNN-BASED HYBRID MODEL

Our proposed approach in the hybrid-based deep learning methods for shilling attack detection is presented in this section. Combining several prediction methods can reduce the weaknesses of single-based deep learning. Besides, hybrid models integrate the benefits of several approaches to increase prediction precision and accuracy. Various hybrid models can be applied to the shilling attack problem. The proposed CNN-based hybrid architecture integrates the advantages of CNN and other models. The self-learning and flexible aspect of CNN makes it a perfect match for extracting features of multidimensional user ratings' profile [18]. Figure 1 depicts the general structure of our proposed hybrid models—Figures 2 to 4 present more details on each hybrid model. At first, user rating profiles are converted into matrices through a preprocessing layer which is the vital process to fulfill the primary requirement of CNN. Further, the CNN layer is used to extract features. Convolutional layers operate on the user-item matrix to reduce

its dimensionality, and then the pooling layer is added to decrease the representation. A data preprocessing layer is used in all the proposed hybrid models to add more flexibility to our end-to-end solution. First attacks are injected into the benchmark dataset (details are discussed in section V). Second, if the input item vectors in the user-item matrix are not our standard rectangle form, a padding method is used to convert them into proper dimensions. Similar to the padding method used in [25], items with zero rates are added to the user-item matrix so that the item vector can be transformed to the rectangle form. The pseudo of the general CNN-based hybrid algorithm is shown in algorithm 1.

Algorithm1: CNN-based hybrid model
Input: User-Item Matrix ratings
Output: Users labeled as genuine or attacker.
Begin
Stage 1- Preprocessing: For all users in the user-item matrix, reshape the array of items to fit into the convolution layer.
Stage 2- CNN-feature extraction: Calculate convolution between reshaped items and a kernel size of (3,3), apply RELU activation function, max pooling layer of (3,3), and the final RELU activation function.
Stage 3- Prediction: Run the classification on the user-item matrix with reduced dimensions and classify users into genuine and attackers.
Stage 4- validation: Calculating validation measures.
End

A. Hybrid CNN+LSTM

A CNN model is first applied to the preprocessed user-item matrix. Then its output is conveyed into the LSTM model, which ties the extracted features from the CNN network and classifies users into attackers and genuine users. Figure 2 shows the structure of this model. The two-dimensional convolutional layer with 32 hidden neurons and a kernel size of 3*3 and RELU activation function is applied on the preprocessed user-item matrix, followed by a Max pooling layer of size 3*3. Dropout regularization is utilized to reduce the number of neurons and avoid overfitting. The output of this is transferred to an LSTM layer with 32 hidden neurons and then goes through a dense layer of 16 units and RELU activation function, then dropout with a rate of 0.5 and ultimately a dense output layer of 2. The hybrid model is run in 5 epochs and 20 batch sizes.

B. Hybrid CNN+GRU

To combine CNN with GRU, we adopt the similar strategy described in part A to detect attack users. The only difference is inside the LSTM and GRU cells. The update and reset gates inside a GRU cell solve the vanishing gradient problem in a standard RNN [20]. Due to the similarity between the network structures of CNN+LSTM and CNN+ GRU models, Figure 2 shows the structure of both as CNN+RNN hybrid architecture. The rating matrix is related ratings of M users to N items in an RS. This model is also run in 5 epochs and 20 batch sizes. Since both LSTM and GRU are types of RNN, algorithm 2 shows the general pseudo code of them both.

C. Hybrid CNN+XGB

Although XGBoost works well on its own and has achieved high performance in many areas, this model is still indefinite for feature learning [25]. The CNN network we have adopted in this model differs from traditional CNN because there is no fully connected (FC) layer. Therefore, there is no need to backward the weights from FC to former layers and re-adjust them, which simplifies the structure.

Algorithm2: CNN-RNN hybrid model
Input: User-Item Matrix ratings
Output: Users labeled as genuine or attacker.
Begin
Stage 1- Preprocessing: For all users in the user-item matrix, reshape the array of items to fit into the convolution layer.
Stage 2- Divide the data into train and test sets with 70% as train and 30% test.
Stage 3- CNN-feature extraction: Calculate convolution between reshaped matrix and a kernel size of (3,3). Apply RELU activation function, max pooling layer of (3,3), and the final RELU activation function.
Stage 4- Prediction**:** Convey the matrix with reduced dimensions to a RNN layer with 32 hidden neurons. Send the results through a dense layer of 16 units and RELU activation function. Apply a dropout layer of 0.5 and a SoftMax activation function at the end to classify users into genuine and attacker.
Stage 5- validation: Calculating validation measures.
End

Another benefit of the CNN+XGB model is that the number of calculation parameters will be reduced in this hybrid model. The CNN network consists of a two-dimensional convolutional layer with a kernel size of 3*3, a RELU activation function, a max-pooling layer of size 3*3, and another RELU activation function at the end. A user-item rating matrix is given as the input of this network. After the key features are learned from the training dataset, the output is passed to the XGBoost model for classification. This architecture is depicted in Figure 3. Algorithm 3 shows the pseudo code of this hybrid model.

Algorithm3: CNN-XGB hybrid model
Input: User-Item Matrix ratings
Output: Users labeled as genuine or attacker.
Begin
Stage 1- Preprocessing: For all users in the user-item matrix, reshape the array of items to fit into the convolution layer.
Stage 2- Divide the data into train and test sets with 70% as train and 30% test.
Stage 3- CNN-feature extraction: Calculate convolution between reshaped matrix and a kernel size of (3,3). Apply RELU activation function, max pooling layer of (3,3), and the final RELU activation function.
Stage 4- Prediction**:** Convey the matrix with reduced dimensions to the XGB model with 100 estimators. Classify users into genuine and attacker using a binary logistic objective.
Stage 5- validation: Calculating validation measures.
End

D. Hybrid CNN+SNN

The proposed hybrid model (Shown in Figure 4) in this section uses a CNN network for feature extraction purposes, similar to the proposed hybrid CNN+XGB model. The CNN network consists of a two-dimensional convolutional layer with a kernel size of 3*3, a RELU activation function, a max-pooling layer of size 3*3. Then the output tensor is fed into the Poisson encoding process to generate spikes.

Algorithm4: CNN-SNN hybrid model

Input: User-Item Matrix ratings
Output: Users labeled as genuine or attacker.
Begin
Stage 1- Preprocessing: For all users in the user-item matrix, reshape the array of items to fit into the convolution layer.
Stage 2- Divide the data into train and test sets with 70% as train and 30% test.
Stage 3- CNN-feature extraction: Calculate convolution between reshaped matrix and a kernel size of (3,3). Apply RELU activation function, max pooling layer of (3,3), and the final RELU activation function.
Stage 4- Spiking neuron layer: Encode the matrix with reduced dimensions into spikes using Poisson encoding. Record Voltage for excitatory and inhibitory layers. Assign 100 neurons and run the spike simulation for 250 milli seconds in each iteration.
Stage 5- Prediction: Update the simulation results in intervals of 10. Assign labels to excitatory layer neurons. Classify users into genuine and attacker.
Stage 6- validation: Calculating validation measures.
End

V. F-DIVERGENCE

In this section, we propose a variant of the F1-score measure to find the similarity between two individual models in an aggregate. This metric, named the F-divergence score, indicates the similarity between predicted labels of each individual model, which helps decide whether to combine them as a hybrid model or not, i.e., the aggregate is a valid ensemble. In the F-divergence calculation, TP' = the number of true positive predicted labels of the CNN model classified by the second model. FN' = The number of false-negative predicted labels of the CNN model classified by the second model. F-divergence is calculated as in Eq.1. The Higher the F-divergence, the close solution both CNN and other models have in the same hybrid.

$$\mathrm{F-divergence} = \frac{TP'}{TP' + \frac{1}{2}(FP' + FN')} \qquad (1)$$

I. EXPERIMENTAL METHOD AND ANALYSIS

This section is divided into three parts to describe the dataset and attack injection process, compare the performance of the proposed models and describe our new proposed compatibility metric. All experiments are done on the Intel Core i5 processor with a 1.60 GHz speed and 8 GB RAM. Except for SNN, all the models are created using the Keras library in Python. SNN model is built in Bindsnet [23] as a simulating SNN package.

A. Dataset and preprocessing

Two benchmark datasets, Movie-Lens 100K and Netflix are used in this paper. In the Movie-Lens dataset, there are 943 users with 100,000 ratings on 1682 movies. Ratings are from 1 to 5, with 5 indicating the best movie. The Netflix dataset has 470,758 users. Since the Movie-Lens dataset is too small compared to the Netflix dataset, a random subset of the Netflix dataset is used in this experiment. It consists of 639 users' rate on 2123 movies. Both datasets are split to train and test, considering 30% as test data. Attacks are injected on four attack types: random, average, bandwagon, average over popular (AOP), with five filler sizes of 1%, 3%, 5%, 7%, 9%, and three attack sizes of 10%, 15% and 20%, similar to the work in [14].

B. Comparative analysis

The performance of four individual models, LSTM, GRU, XGBoost, SNN, and four proposed hybrid models CNN-LSTM, CNN-GRU, CNN-XGB, and CNN-SNN, are compared based on the F1-score. Each model is run for 60 different scenarios on each dataset, and the results are calculated as the average performance per parameter. Based on the results, the deep learning method has achieved high scores on all attack types, unlike traditional methods, which perform poorly on more obfuscated attacks such as AOP. As shown in Figures 5 to 10, the hybrid model outperforms the individual model in most scenarios. For instance, the hybrid CNN-XGB model has improved the performance in the Netflix dataset, while hybrid CNN-SNN has resulted in a lower F1-score. Therefore, we decide whether combining two individual models will improve the result or not. Otherwise, it may end up consuming computational power without adding any improvement to the model performance. As shown in Table 1, the F-divergence values of all hybrid models except CNN-SNN are above 93%. This shows these models are predicting the attacks primarily similar. Only CNN-SNN has lower F-divergence results. The reason is that the SNN model shows its best result with under-sampling data. To compare it with CNN, the same under-sampling method is applied to the CNN model, which has deteriorated its performance. Therefore, their results are not very similar and show a lower F-divergence score. And the hybrid model does not perform better than single CNN or SNN. But the rest of the hybrid models with high F-divergence scores (more than 90%) show good performance on the hybrid model but not necessarily better than both single models. Thus, we have broken down the F-divergence into buckets to compare the number of cases where the hybrid model has shown better performance than single models. The number of better hybrid result columns in Tables 1 and 2 indicates the number of test scenarios where the F1-score of the hybrid model is higher than both individual models. We can observe that when combining two models if both individual models achieve high performance, the higher their F-divergence score, the better the average F-score of the hybrid model will be. This is derived from calculating the likelihood of a hybrid model to show a higher F-score than each individual model based on 240 observations on each benchmark dataset. Applying the same bucketing approach on the Movie Lens results, as shown in Table 2, does not show any specific trend, and the number of cases with better hybrid results is so low.

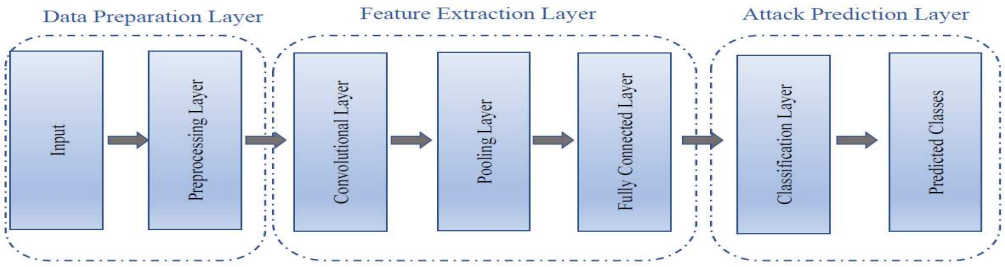

Figure 1. The CNN–based hybrid architecture

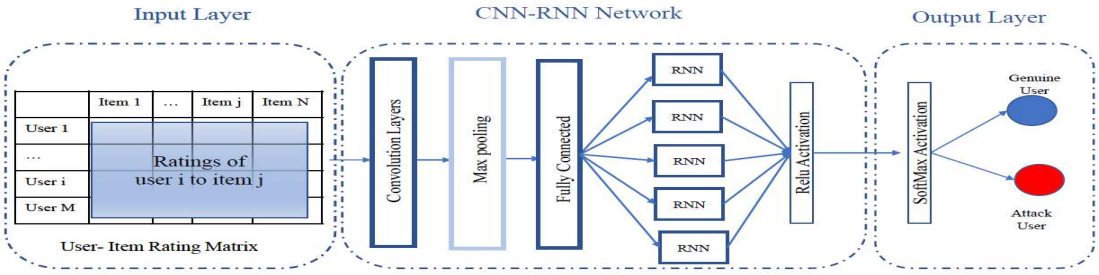

Figure 2. The CNN–RNN architecture

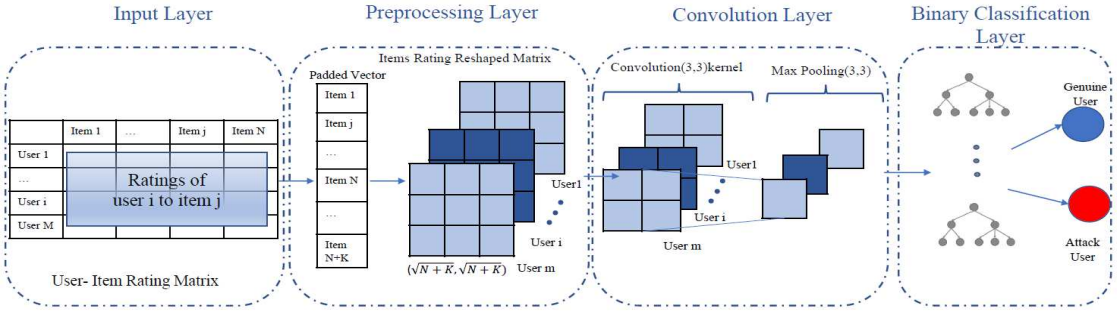

Figure 3. The CNN–XGB architecture

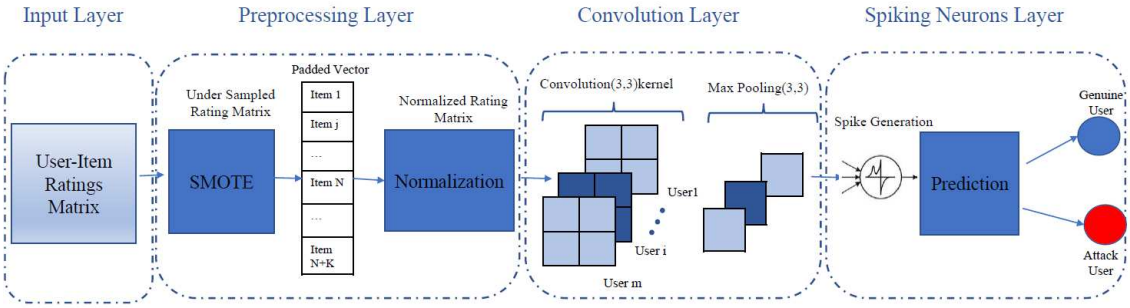

Figure 4. The Proposed CNN–SNN architecture

TABLE I. F-DIVERGENCE (NETFLIX DATASET)

Buckets of F-divergence score	# of better hybrid result	Total number of test scenarios	Probability of getting a better result from combining two models	Average F1-Score of First Model	Average F1-Score of Second Model	Average of F-Divergence Score	Average F1-score of hybrid Model
> 93%	106	180	0.589	0.88	0.90	0.97	0.90
> 94%	105	177	0.593	0.88	0.90	0.97	0.90
> 95%	99	168	0.589	0.88	0.90	0.97	0.90
> 96%	88	149	0.591	0.89	0.91	0.97	0.91
> 97%	71	118	0.602	0.89	0.91	0.98	0.91
> 98%	44	68	0.647	0.91	0.92	0.98	0.92
> 99%	11	18	0.611	0.91	0.93	0.99	0.92

TABLE II. F-DIVERGENCE (MOVIE-LENS DATASET)

Buckets of F-divergence score	# of better hybrid result	Total number of test scenarios	Probability a better result from combining two models	Average F1-Score of First Model	Average F1-Score of Second Model	Average of F-Divergence Score	Average F1-score of hybrid Model
> 93%	34	212	0.16	0.98	0.99	0.98	0.96
> 94%	34	207	0.16	0.98	0.99	0.98	0.96
> 95%	34	205	0.17	0.98	0.99	0.98	0.96
> 96%	34	204	0.17	0.98	0.99	0.99	0.96
> 97%	30	192	0.16	0.98	0.99	0.99	0.96
> 98%	22	147	0.15	0.98	0.99	0.99	0.97
> 99%	11	60	0.18	0.99	1.00	1.00	0.98

The main reason for this is that the number of instances where hybrid results showed better performance than single models is much less in the Movie Lens dataset in comparison with Netflix Dataset. Due to the different sparsity levels of these datasets, we can see that single models already show good performance on Movie Lens (less sparse dataset). With high values of F-divergence, the combination of the models does not add much value. Although there are more cases with high F-divergence scores in the Movie-Lens dataset because of the high F-scores of single models, the hybrid model's F-score does not show significant improvement. Table 3 shows the F-divergence values for each dataset with the sparsity levels. We can see that the Netflix dataset shows more sparsity on the User-Item matrix. The individual CNN model on the Netflix dataset has not performed as well as it did on Movie-Lens. Therefore, using hybrid models has resulted in a larger number of cases in which the hybrid result was better than both individual CNN and adopted model. 106 cases out of 240 have resulted in higher performance using hybrid models. 240 total tested cases are related to four hybrid models (CNN-LSTM, CNN-GRU, CNN-SNN, and CNN-XGB) using three different attack sizes (10%, 15%, 20%) and five filler sizes (1%, 3%, 5%, 7%, 9%) and four attacks (random, average, Bandwagon and AOP). Figures 5 and 6 represent the sensitivity of F1-score to attack types for each model on Movie Lens and Netflix dataset respectively. We can see that models have performed well on different attack types and even obfuscated attacks such as AOP. Figures 7 and 8 show the sensitivity of the average F1-score to the attack sizes of 10%, 15% and 20% of all items. It shows the models' performance is robust on different variations of attack sizes. XGB and CNN-XGB models show the highest F1-score compared to the other models while CNN-SNN shows the lowest prediction score specially on the Netflix dataset. On average, the F1-score of the hybrid model on Netflix is not high due to the impact of the low performance of the CNN-SNN model. Figures 9 and 10 depict the average F1-score of models based on five different filler sizes. The filler size increases the F1-score deteriorates.

I. CONCLUSION AND FUTURE DIRECTIONS

In this paper, new models and architectures are proposed for hybrid learning methods to detect shilling attacks in RS. Four individual models, including LSTM, GRU, XGBoost, and SNN, are adopted to classify users into genuine and attack users, and four hybrid CNN-based models are proposed and compared together. We have used the CNN layers to extract features from the user-item matrix, and the second model is used for classification. We have proposed a new compatibility metric to assess the validation of the hybrid model before an aggerate is designed. The proposed measure provides a quantitative assessment tool that saves computational time and efforts of building a redundant aggregate. We have concluded that combining two models with good performance and a high F-divergence score results in a higher F-score for the hybrid model when the sparsity level is high. Future directions would include incorporating various statistical measures in addition to the F-divergence. In addition, further investigation of the hybrid machine learning models is needed.

TABLE III. F-DIVERGENCE RESULTS.

Data Set	MovieLens	Netflix
User-Item matrix Sparsity Level	93%	96%
# of cases with better hybrid result	39	106
Total tested cases	240	240
Average F1- Score of CNN Model	0.96	0.84
Average F1- Score of Second Model	0.99	0.86
Average of F- Divergence Score	0.97	0.89
Average F1- Score of hybrid Model	0.94	0.74
Max F1 -Score of hybrid Model	1	0.962593

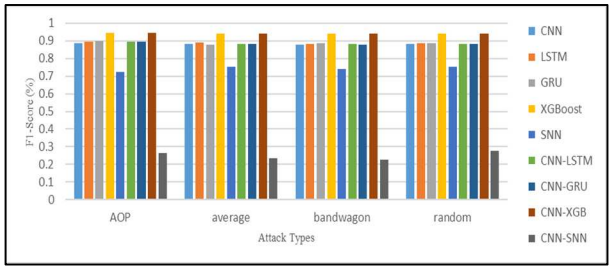

Figure 5. F1-Score vs. Attack types on MovieLens dataset

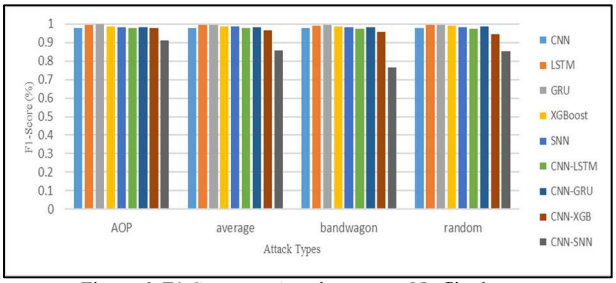

Figure 6. F1-Score vs. Attack types on Netflix dataset

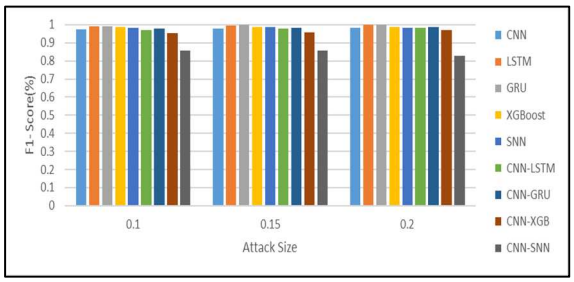

Figure 7. F1-Score vs. Attack Sizes on MovieLens dataset

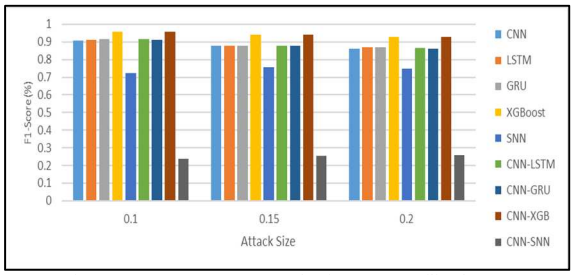

Figure 8. F1-Score vs. Attack Sizes on Netflix dataset

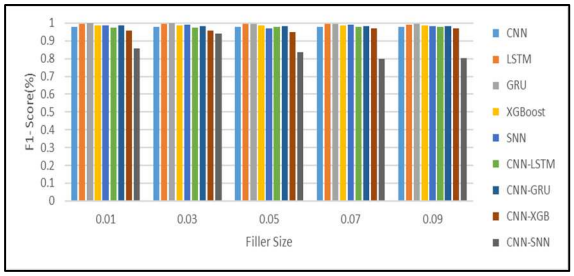

Figure 9. F1-Score vs. Filler Sizes on MovieLens dataset

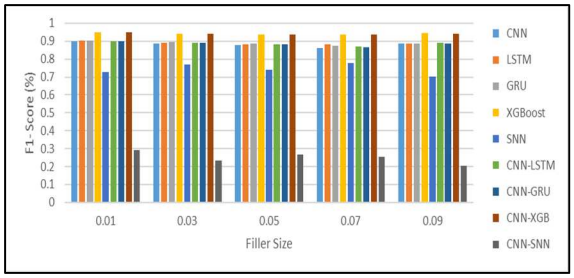

Figure 10. F1-Score vs. Filler Sizes on Netflix dataset

REFERENCES

[1] Chirita, P.-A.; Nejdl, W.; Zamfir, C. Preventing shilling attacks in online recommender systems. In Proceedings of the 7th ACM International Workshop on Web Information and Data Management, 4, 2005; 67–74.

[2] Xia, H.; Fang, B.; Gao, M.; Ma, H.; Tang, Y.; Wen, J. A novel item anomaly detection approach against shilling attacks in collaborative recommendation systems using the dynamic time interval segmentation technique. Inf. Sci. 2015, 306, 150–165.

[3] Yang, Z.; Cai, Z. Detecting Anomalous Ratings in Collaborative Filtering Recommender Systems. Int. J. Digit. Crime Forensics 2016, 8, 16–26.

[4] Bilge, A.; Ozdemir, Z.; Polat, H. A Novel Shilling Attack Detection Method. Procedia Comput. Sci. 2014, 31, 165–174.

[5] Zayed, R.A.; Ibrahim, L.F.; Hefny, H.A.; Salman, H.A. Shilling Attacks Detection in Collaborative Recommender System: Challenges and Promise. In Workshops of the Int. Conf. on Advanced Information Networking and Applications, 2020; Springer:; pp. 429–439.

[6] Zhang, S.; Chakrabarti, A.; Ford, J.; Makedon, F. Attack detection in time series for recommender systems. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia PA, USA, 20–23th August 2006; Association for Computing Machinery: New York NY, USA; pp. 809–814.

[7] Lee, J.-S.; Zhu, D. Shilling Attack Detection—A New Approach for a Trustworthy Recommender System. INFORMS J. Comput. 2012, 24.

[8] Alonso, S.; Bobadilla, J.; Ortega, F.; Moya, R. Robust Model-Based Reliability Approach to Tackle Shilling Attacks in Collaborative Filtering Recommender Systems. IEEE Access 2019, 7, 41782–41798.

[9] Mehta, B.; Nejdl, W. Unsupervised strategies for shilling detection and robust collaborative filtering. User Model. User Adapt. 2008, 19, 65–97

[10] Tong, C.; Yin, X.; Li, J.; Zhu, T.; Lv, R.; Sun, L.; Rodrigues, J.J.P.C. A shilling attack detector based on convolutional neural network for collaborative recommender system in social aware network. Comput. J. 2018, 61, 949–958, doi:10.1093/comjnl/bxy008.

[11] Ebrahimian, M. and Kashef, R., 2020, December. Efficient Detection of Shilling's Attacks in Collaborative Filtering Recommendation Systems Using Deep Learning Models. In *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 460-464.

[12] Zhou, Q.; Wu, J.; Duan, L. Recommendation attack detection based on deep learning. J. Inf. Secur. Appl. 2020, 52, 102493.

[13] Gao, J.; Qi, L.; Huang, H.; Sha, C. Shilling Attack Detection Scheme in Collaborative Filtering Recommendation System Based on Recurrent Neural Network. In Future of Information and Communication Conference, San Francisco CA, USA, 5–6th March 2020; Springer: Cham, Switzerland; pp. 634–644.

[14] Ebrahimian, Mahsa, and Rasha Kashef. "Detecting Shilling Attacks Using Hybrid Deep Learning Models." Symmetry 12, no. 11 (2020): 1805.

[15] Rani, S., Kaur, M., Kumar, M., Ravi, V., Ghosh, U. and Mohanty, J.R., 2021. Detection of shilling attack in recommender system for YouTube video statistics using machine learning techniques. *Soft Computing*, 1-13.

[16] Hao, Y. and Zhang, F., 2021. An unsupervised detection method for shilling attacks based on deep learning and community detection. *Soft Computing*, *25*(1), pp.477-494.

[17] Wu, F., Gao, M., Yu, J., Wang, Z., Liu, K. and Wange, X., 2021. Ready for Emerging Threats to Recommender Systems? A Graph Convolution-based Generative Shilling Attack. *arXiv preprint arXiv:2107.10457*.

[18] Vivekanandan, K.; Praveena, N. Hybrid convolutional neural network (CNN) and long-short term memory (LSTM) based deep learning model for detecting shilling attack in the social-aware network. J. Ambient. Intell. Humaniz. Comput. 2020, 0123456789, 1–14.

[19] Alec Yenter and Abhishek Verma.Deep cnn-lstm with combined kernels from multiple branches for imdb review sentiment analysis. IEEE 8th Annual Ubiquitous Computing, Electronics and MobileCommunication Conference (UEMCON), pages 540–546., 2017.

[20] Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv 2014.

[21] T. Chen, C. Guestrin, Xgboost: A Scalable Tree Boosting System, 2016. ArXiv e- printsarXiv:1603.02754.

[22] Boudjelal Meftah, Olivier L´ezoray, Soni Chaturvedi, Aleefia A Khurshid, and Abdelkader Benyettou. Image processing with spiking neuron networks. In Artificial Intelligence, Evolutionary Computing and Metaheuristics, pages 525– 544. Springer, 2013.

[23] Hazan, Hananel, Daniel Saunders, Darpan T. Sanghavi, Hava Siegelmann, and Robert Kozma. "Unsupervised learning with self-organizing spiking neural networks." In 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1-6. IEEE, 2018.

[24] Agebure, M.A., Baagyere, E.Y. and Oyetunji, E.O., 2020. Spiking Neural Network Learning Models for Spike Sequence Learning and Data Classification. *Asian Journal of Research in Computer Science*, pp.1-17.

[25] Thongsuwan, S., Jaiyen, S., Padcharoen, A. and Agarwal, P., 2021. ConvXGB: A new deep learning model for classification problems based on CNN and XGBoost. *Nuclear Engineering and Technology*, *53*(2), pp.522-531.