Phishcasting: Deep Learning for Time Series Forecasting of Phishing Attacks

Syed Hasan Amin Mahmood
Department of Electrical Engineering
Lahore University of Management Sciences
Lahore, Pakistan
syed.mahmood@lums.edu.pk

Syed Mustafa Ali Abbasi
Department of Computer Science
Lahore University of Management Sciences
Lahore, Pakistan
20100029@lums.edu.pk

Ahmed Abbasi
Department of IT, Analytics, and Operations
University of Notre Dame
Notre Dame, IN, USA
aabbasi@nd.edu

Fareed Zaffar
Department of Computer Science
Lahore University of Management Sciences
Lahore, Pakistan
fareed.zaffar@lums.edu.pk

***Abstract*—Phishing attacks remain pervasive and continue to be a source of significant monetary loss, identity theft, and malware. One of the challenges is that in most organizational settings, the detection paradigm is inherently about identifying and reacting to threats in real-time, as they are unfolding. As a way to complement these efforts with greater foresight, we introduce the idea of phishcasting — forecasting of phishing threat levels weeks or months into the future. Given that phishing attack volume time series data is noisy and devoid of traditional seasonal and cyclical trends, we extend the time series forecasting framework to utilize multiple time series, auxiliary information and alternate representations. We also introduce CoT-Net, a flexible, end-to-end CNN-LSTM based deep learning method for forecasting of complex phishing attack volume time series. CoT-Net uses time series embeddings to uncover correlations between organizational attack patterns within and across industry sectors. Using a publicly available test bed featuring multiple organizations' attack volume over time, we find CoT-Net to outperform most state-of-the-art time series forecasting methods. By showing that phishcasting might be possible and practical, our work has important proactive implications for cybersecurity.**

***Index Terms*—Phishing, forecast, deep learning, cybersecurity, time series, predict, machine learning.**

I. Introduction

Phishing remains one of the most prevalent forms of cybercrime, with nearly half a million unique phishing attacks reported, and almost twice as many phishing websites detected in 2019 alone [1]. Phishing attacks not only defraud individuals and organizations of millions of dollars but also pose additional enterprise-level security threats, including identity theft and viruses [2], [3]. While significant progress has been made over the past two decades to prevent users from falling prey to phishing [4], the problem remains far from resolved.

Given that phishing is a type of semantic attack that exploits human vulnerabilities as opposed to those in hardware and software [3], it makes sense that the majority of anti-phishing research to date has examined socio-technical phenomena related to mitigating damage from phishing attacks as they are unfolding [5]. One large stream of work has examined user susceptibility to phishing —- what factors make some users more likely to fall prey to phishing attacks [6], [7], and how to alleviate susceptibility [8]–[10]. Another has explored machine learning methods for detecting phishing threats [11], [12]. This work has provided a wealth of insights and best practices for the point of attack and the importance of broader training and mindfulness initiatives [13]. Still, what if we could complement these crucial existing organizational security practices with longer term forecasts of phishing threat levels? An idea we call *phishcasting*. If accurate, such forecasts about the level of attack volumes over the coming days, weeks, or even months could allow organizations to plan and prepare in a more agile, efficient, and proactive manner.

Time series forecasting has historically been an important area of research since many real world phenomena can benefit from proactive foresight [14]. In this work, we explore time series modeling of organizations' phishing attack volumes. The two key contributions of our paper are as follows:

- To the best of our knowledge, this is the first study to analyze phishing through the forecasting lens. We investigate phishcasting as a noisy time series modeling problem. As part of this effort, we systematically leverage auxiliary information and alternate representations, including hierarchical industry-organizational structures.
- We propose CoT-Net, a flexible deep learning model for forecasting of complex, correlated time series in an end-to-end fashion using time series embeddings in conjunction with CNN and LSTM models. We demonstrate the viability of CoT-Net through extensive evaluation on longitudinal phishing attack volume data related to several organizations in the technology and finance sectors.

II. Related Work

We believe this paper presents one of the first-of-its-kind investigations into the phishcasting problem. More broadly,

work pertaining to time series forecasting in cybersecurity contexts has been limited. Some recent studies utilized social media mentions as a signal to predict cyberattacks [15], including coupling Twitter data and neural networks to forecast Distributed Denial of Service (DDoS) attacks [16]. [17] proposed a deep neural network to differentiate cyberattack behaviors and predict future attacks. However, due to the noisy and complex nature of cyberattacks, these studies define and model predictions as less precise future outcomes. Conversely, [18] investigates the use of a traditional ARIMA forecasting model to predict future number of attacks at the network layer. [19] applies deep learning-based forecasting models to simulated synthetic cyber attack data as an intermediate step for the industrial fault detection problem.

Modern machine learning methods provide a means to learn temporal dynamics in a primarily data-driven manner [20]. Commonly used time series models inspired by statistics and econometrics are ARIMA and Prophet. ARIMA performs time series forecasting on a stationary time series by regressing the value at the current time step on p lagged values and q lagged forecast errors. The stationary time series is created by performing d-th order differencing on the original series. Prophet by Facebook is an additive regression model with four main components: linear or logistic growth curve trend, seasonality modeled using Fourier series and dummy variables, user-provided list of holidays, and error term assumed to be normally distributed. Classic machine learning methods utilizing windowed data matrices, including tree-base classifiers, support vector machines (SVMs), and regression have also been used extensively on an array of problems [20], [21].

In recent years, deep learning techniques such as stateless and stateful long short-term memory networks (LSTMs), as well as 1-D convolution, have garnered attention for time series modeling [20], [21]. Example applications include deep learning for electric load [22] and influenza prevalence forecasting [23]. Although a large number of deep learning models have been developed for time series forecasting, various limitations and opportunities still exist [20]. First, many existing methods are highly contextualized to their respective solution areas [22], [23]. Second, significant research focuses on synthetic or real datasets that follow neat properties of stationarity, cyclicality, seasonality, completeness and/or non-sparsity — their effectiveness on noisy, *complex* datasets devoid of such properties remains unclear [24]. Third, phishing forecasting entails consideration for the hierarchical structure of time series with logical groupings between trajectories, for example companies that have common phishing attack volume trends. As we later demonstrate empirically in the evaluation section, due to these three limitations or gaps, existing ARIMA, classic machine learning, and standard deep learning methods are not as effective for forecasting phishing attack volumes over time.

III. Phishcasting Problem Formulation

In this section, we describe our formulation for the phishing attack volume time series forecasting framework which leverages multiple (correlated) time series, auxiliary information and alternate representations for accurate forecasting of complex time series data. The proposed framework is applicable to both regression and classification tasks.

We assume to have K time series of length T each, which we represent as a matrix $\mathcal{D} \in \mathbb{R}^{T \times K}$, element d_{tk} of which corresponds to the value at time unit t of time series k. We may also have access to auxiliary information, such as date and domain-specific features, associated with each element of $\mathcal{D}$. Given this data, our objective is to obtain accurate forecasts of time series r, which we refer to as the target time series, where $r \in \{k \in \mathbb{Z}^+ \mid k \leq K\}$.

$K' \subseteq \{k \in \mathbb{Z}^+ \mid k \leq K\}$ is the set of time series of high relevance to our forecasting problem. τ denotes the lag length or the number of past terms to include in one data point. ϕ denotes the step interval. For simplicity, it also refers to how far ahead we want to forecast (i.e., our prediction horizon is ϕ future time intervals). $T' = \{t' \in \tau + \lambda\phi \mid \lambda \in \mathbb{N}_0 \wedge t' \leq T\}$ consists of the last time unit for each data point, and therefore $|T'| = \lfloor \frac{(T-\tau)}{\phi} \rfloor$ is the total number of data points.

The input $\mathbf{X}$ under our framework comprises some or all of the following: lag features ($\mathbf{L}$), auxiliary features ($\mathbf{A}$) and embedding features ($\mathbf{E}$).

1) **Lag features** refer to specific values from the past. $\mathbf{l}_{tk}$ is the lag feature vector containing τ values from time unit $(t - \tau + 1)$ to t for time series k.

$$\mathbf{l}_{tk} = [d_{(t-\tau+1)k}, d_{(t-\tau+2)k}, \ldots, d_{tk}]$$

We obtain a matrix $\mathbf{L}_t \in \mathbb{R}^{|K'| \times \tau}$ by stacking vertically the vectors $\mathbf{l}_{tk'}\ \forall k' \in K'$. We compute $\mathbf{L}_{t'}\ \forall t' \in T'$ and stack them together to obtain the main lag features matrix $\mathbf{L} \in \mathbb{R}^{|T'| \times |K'| \times \tau}$. Fig. 1 provides an illustration for the creation of lag features matrix $\mathbf{L}$.

2) **Auxiliary features** refer to structured information related to time series data. Such information, when available, takes the form of $\mathbf{a}_{tk} \in \mathbb{R}^n$ associated with each element d_{tk} of $\mathcal{D}$. We obtain a matrix $\mathbf{A}_t \in \mathbb{R}^{|K'| \times n}$ by stacking vertically the vectors $\mathbf{a}_{tk'}\ \forall k' \in K'$. We compute $\mathbf{A}_{t'}\ \forall t' \in T'$ and stack them together to obtain the main auxiliary features matrix $\mathbf{A} \in \mathbb{R}^{|T'| \times |K'| \times n}$.

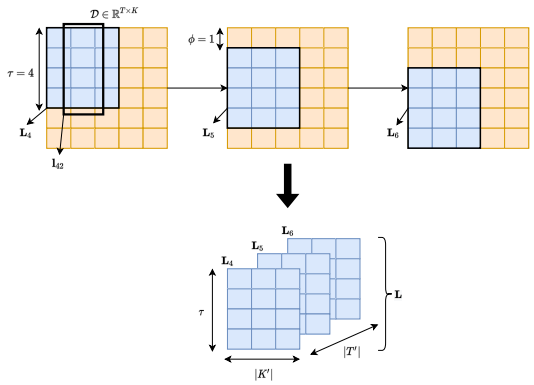


Fig. 1: Illustration for lag features matrix $\mathbf{L}$ formation. $K = 5$, $K' = \{1, 2, 3\}$, $T = 6$ and $T' = \{4, 5, 6\}$ here.

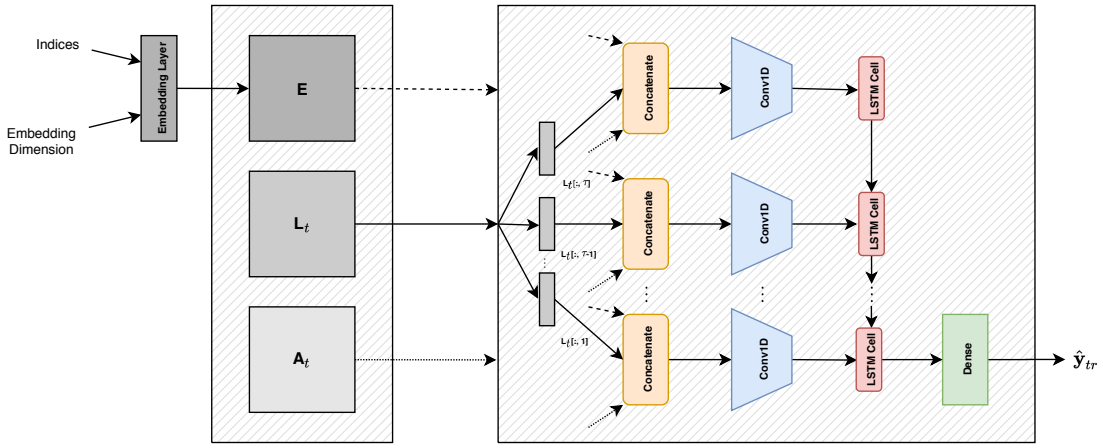


Fig. 2: Illustration of CoT-Net, the proposed deep learning model, at time step t.

3) **Embedding features** — or simply embeddings — refer to alternate representation of time series data. They can be provided offline or learnt online. The construct is generic and flexible. Let $\mathbf{e}_k \in \mathbb{R}^m$ be embedding for time series k. We stack $\mathbf{e}_{k'}\ \forall k' \in K'$ to obtain the main embeddings matrix $\mathbf{E} \in \mathbb{R}^{|K'| \times m}$.

The target value $\mathbf{y}$ depends on the problem type — future predicted volumes for regression, and binary prediction trend for classification. $\mathbf{y} = \{\mathbf{y}_{tr} \mid t \in T'\}$ in our problem setup. The framework, however, is easily extensible to joint forecasting of all time series, i.e., where $\mathbf{y} = \{\mathbf{y}_{tk} \mid t \in T' \wedge k \in K'\}$.

Having defined the input and the target value, our goal is to find a predictor function $h_\theta : \mathcal{X}_t \mapsto \mathcal{Y}_{tr}$, where $h_\theta \in \mathcal{H}$, the hypothesis space, such that $\hat{\mathbf{y}}_{tr} = h_\theta(\mathbf{X}_t)$ is a good approximation of $\mathbf{y}_{tr}$. Note that $\mathbf{X}_t \subseteq \{\mathbf{L}_t, \mathbf{A}_t, \mathbf{E}\}$ and θ is the set of parameters of the predictor function. We also define a loss function $\mathcal{L} : \mathcal{Y}_{tr} \times \mathcal{Y}_{tr} \mapsto \mathbb{R}^+$ to measure the performance of our predictor. Our objective then is to find the optimal set of parameters such that the average loss is minimized.

$$\theta = \underset{\theta'}{\operatorname{argmin}} \frac{1}{|T'|} \sum_{t \in T'} \mathcal{L}(h_{\theta'}(\mathbf{X}_t), \mathbf{y}_{tr})$$

Note that this framework reduces to standard time series forecasting when $\mathbf{X} = \{\mathbf{L}\}$ and $K' = \{r\}$.

IV. Proposed CoT-Net Architecture

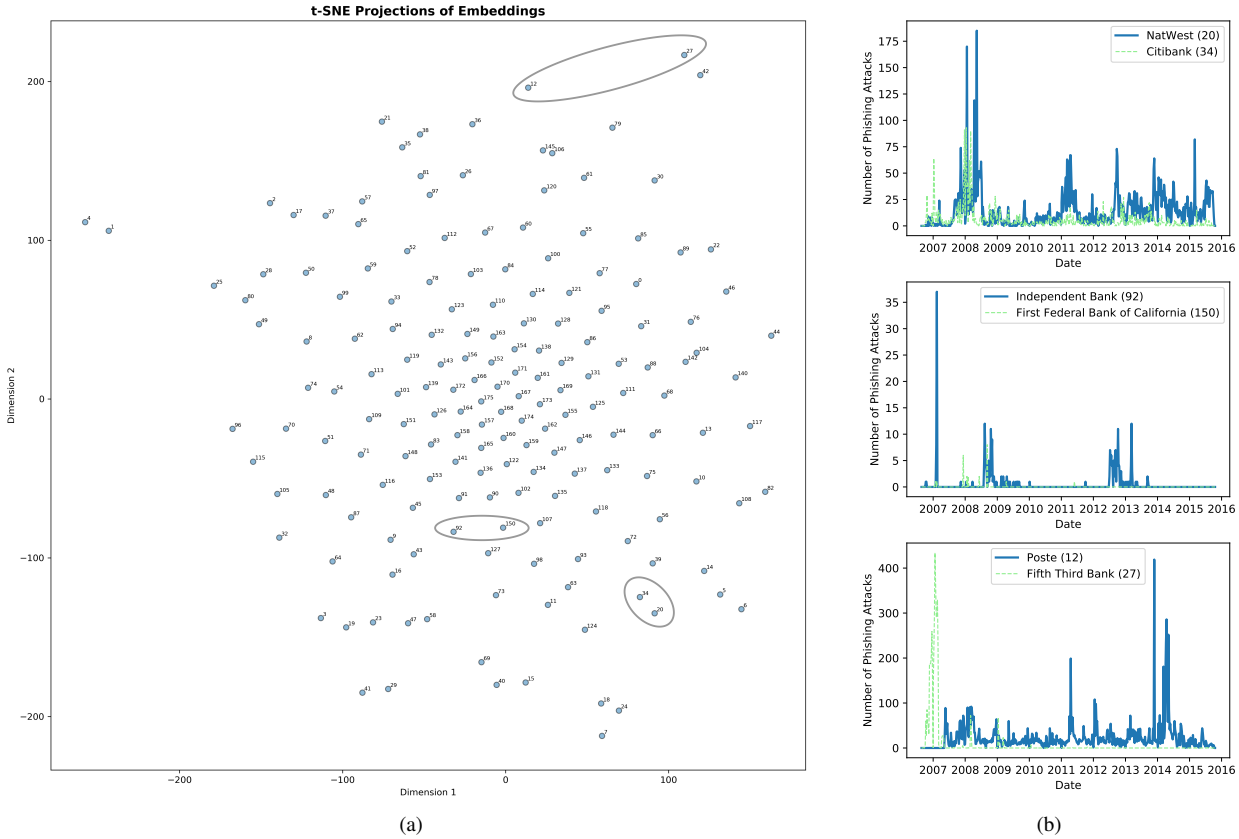
Fig. 2 depicts CoT-Net, our proposed deep neural **net**work for forecasting of **co**mplex, **co**related **t**ime series data, which we use to model the phishcasting problem formulation presented in the prior section. CoT-Net couples 1-D CNNs with stateless LSTMs and time series embeddings to learn local and sequential/temporal patterns in unison. It takes as input the lag features $\mathbf{L}$, auxiliary features $\mathbf{A}$ and embeddings $\mathbf{E}$ (discussed in Section III). Auxiliary features are one-hot encoded. As shown in the figure, concatenation blocks merge lag features and auxiliary features at particular time steps with the embedding features, enabling better end-to-end learning.

An important component of CoT-Net are the time series embeddings. In order to illustrate the intuition and potential predictive benefit of such embeddings, we proposed an offline *Polynomial Embeddings*. A time series can be represented in terms of polynomial coefficients found after polynomial fitting, which can result in a meaningful mechanism for noise removal and fixed, lower dimensional representation of a time series window. We take this concept further and perform 'chunk-wise' polynomial fitting, and use the obtained coefficients to obtain a rich alternate lower dimensional representation (i.e., the Polynomial Embedding) of the original time series. The procedure is more formally outlined in Algorithm 1.

Algorithm 1: Polynomial Embedding

Input : Univariate time series $[x_1, x_2 \dots x_T] \in \mathbb{R}^T$,
Number of Chunks $C \leq T$,
Polynomial Degree d
Output: Polynomial Embedding $\mathbf{e} \in \mathbb{R}^{C(d+1)}$
begin
  $\mathbf{e} \leftarrow [\,]$
  $chunkSize = \lfloor \frac{T}{C} \rfloor$
  $i \leftarrow 1$
  while $i \leq T$ **do**
    $poly \leftarrow \theta_0 + \theta_1 z + \theta_2 z^2 + \ldots + \theta_d z^d$
    $chunk \leftarrow [x_i, x_{i+1}, \ldots, x_{i+chunkSize-1}]$
    $\theta_0, \theta_1, \ldots, \theta_d \leftarrow$
     `FitPolynomial`$(poly, chunk)$
    $\mathbf{e}$.append$(\theta_0, \theta_1, \ldots, \theta_d)$
    $i \leftarrow i + chunkSize$
  end
end

If the embeddings for each organization (i.e., values of the d degree polynomials across chunks) are able to effectively capture key structure, we would expect that time series sharing similar characteristics will have polynomial embeddings that lie close together in the $C(d+1)$-th dimensional space. In the phishcasting context, companies with closer polynomial embeddings are expected to be have similar temporal trends of phishing attacks, and this information can be leveraged to



(a) (b)

Fig. 3: t-SNE projections of Polynomial Embeddings for multiple companies is shown in (a). Actual time series of pairs of companies marked in (a) is provided in (b). Top plot depicts companies very close on the t-SNE plot and they indeed have similar time series trend. Middle plot shows companies somewhat close together and they have some similarities like peaks in close proximity. Bottom plot shows companies relatively far apart and they have rather dissimilar time series.

enhance forecasting. To illustrate this idea, we computed the polynomial embeddings across the time series for nearly 200 companies. Fig. 3 presents a t-SNE plot of embeddings created using 20 chunks C and a polynomial degree d of 20. Each point in the plot signifies a firm's phishing attack volume time series over a ten year period. On the right side of the figure, we overlaid the original time series for three sets of points. Indeed, the firms closer together in the t-SNE plot had time series shapes that looked much more similar, relative to firms further apart on the plot. To counter the rigidity of offline, predefined embeddings, and develop a truly end-to-end framework, in CoT-Net we learn the fixed, lower dimensional representation for each firm using the embedding layer from Keras.

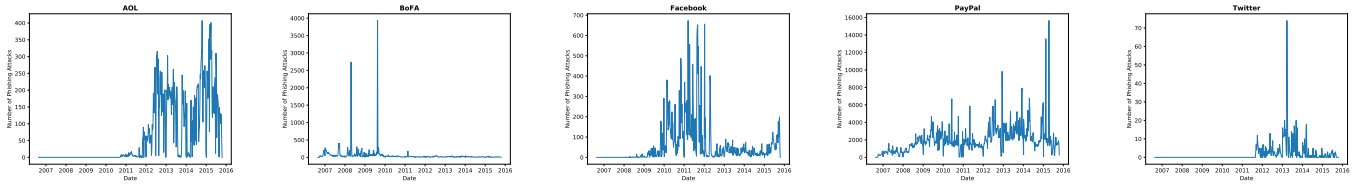
V. Experiments

A. Testbed and Comparison Methods

We test performance across time series data taken from the PhishMonger project repository [25]. This data includes over 1.5 million unique verified phishing attacks related to over 100 targeted websites/brands over a 10 year period from 2006 through 2015. For this study, we focus on five of the most highly targeted brands in the social media, financial services, and internet sectors: AOL, Bank of America, Facebook, PayPal, and Twitter (see Fig. 4). The time series of organizations were standardized to make the input spaces more manageable for the models in both the classification and regression tasks.

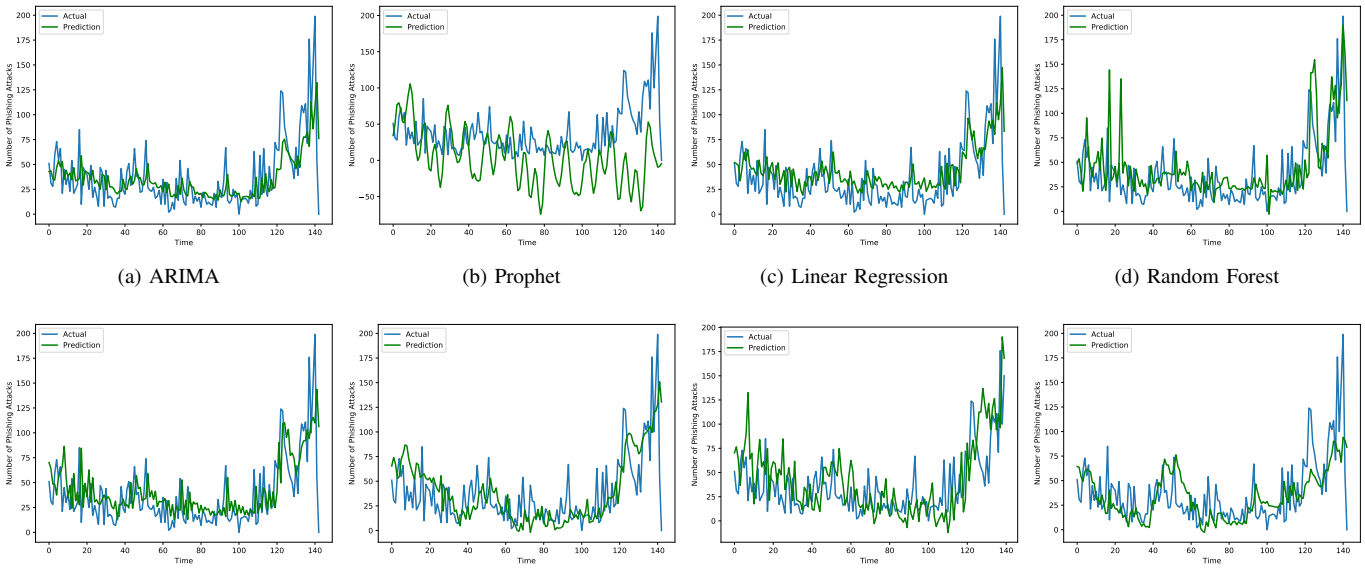
We evaluated our proposed model against several baseline models corresponding to the three types of methods discussed in Section II. The **classic machine learning models** for the regression problem included Linear, Ridge, and Lasso regressions, Random Forest Regressors, and Gradient Boosted Regression Trees (GBRTs). For classification, we used Logistic Regression, Support Vector Machines with Polynomial (SVM-Poly) and Radial Basis Function (SVM-RBF) kernels, Random Forest Classifiers and AdaBoost.

The **standard deep learning models** involved simple multilayer perceptrons (MLPs) and LSTMs. We evaluated both stateful and stateless versions of LSTMs. The models were implemented using Keras. The loss function was adapted based on whether the problem was regression or classification. Built-in Adam optimizer with default parameters was used to train all models. The **other models** examined were ARIMA and Prophet. ARIMA was implemented using the pmdarima library with auto_arima function which automatically selects the best values for p, d and q. Prophet was implemented using the fbprophet library.



(a) AOL (b) Bank of America (c) Facebook (d) PayPal (e) Twitter

Fig. 4: Time series data of companies selected for evaluation and reporting.



(a) ARIMA (b) Prophet (c) Linear Regression (d) Random Forest

(e) Multilayer Perceptron (f) Stateless LSTM (g) Stateful LSTM (h) CoT-Net

Fig. 5: Forecasts of selected models on Facebook data.

B. Evaluation Results

CoT-Net and the comparison methods were applied to the aforementioned 10 years of weekly phishing attack volume data for five companies (see Fig. 4). Regarding our problem formulation matrices, attack timestamp information was used to build the auxiliary features matrix $\mathbf{A}$ by extracting one-hot encoded month and week of year information. We selected K' to be the set of those time series belonging to the same North American Industry Classification System (NAICS) group (e.g., social media or commercial banking). Elements of K' were also used as the indices for the embedding layer to generate $\mathbf{E}$. Input $\mathbf{X}$ was selected to be $\mathbf{L}$ for all models except CoT-Net, where $\mathbf{X}$ was $\{\mathbf{L}, \mathbf{A}, \mathbf{E}\}$.

Consistent with prior time series forecasting studies, we used a sliding window scheme with the first six years used for initial training, and then slid the training window ahead after having tested on the earlier unseen data. Batch retraining was used to update weights of deep learning models since they take longer to train, while full retraining was performed on sliding windows for all other methods. For the regression problem, methods were evaluated using root mean square error (RMSE). For classification, we report mean accuracy, which is essentially the average prediction trend performance.

The results for phishcasting as a regression problem are presented in Table I, whereas the classification results appear in Table II. CoT-Net attained the lowest RMSEs on four of the five firms in the regression setup (the stacked stateless LSTMs had better results on Bank of America). CoT-Net also had the best classification accuracy on three of the five companies. However, the results also shed light on the non-trivial, challenging nature of phishcasting. CoT-Net's prediction trends of 0.62 to 0.66 are still lower than the 0.7 threshold often considered important in other forecasting tasks such as stock movement prediction (although it is markedly better than the below 0.5 values attained by some methods). Further work might be needed to make prediction trend classification more robust and practically valuable. However, the regression RMSE values for firms such as Facebook and Bank of America (two highly phished firms) look promising.

In order to illustrate the practical value of the regression predictions, actual predictions of selected methods are provided in Fig. 5 for visual comparison on the Facebook data. Looking at the figures, CoT-Net and ARIMA appear to be

good at modelling small variations without being 'misled' by volatile, perhaps noisy, data. In constrast, Prophet, stateful LSTMs and others perform poorly. The results for Prophet especially underscore how different phishcasting is from traditional seasonal/cyclical forecasting such as sales prediction. Overall, the results demonstrate the potential (and challenges) for phishcasting as well as some of the design principles embodied in CoT-Net such as the time series embeddings necessary for *local* learning from neighbor firms.

TABLE I: Regression Results (Root Mean Square Error)

	AOL	BofA	Facebook	PayPal	Twitter	AVERAGE
ARIMA	81.24	28.49	25.46	1967.22	8.44	422.17
Facebook Prophet	103.52	70.82	59.32	2039.04	7.78	456.10
Linear Regression	82.20	29.76	26.28	1980.38	9.80	425.68
Lasso Regression	81.51	30.74	26.28	1979.93	8.44	425.38
Ridge Regression	82.16	29.79	26.28	1980.19	9.79	425.64
Random Forest	85.87	13.84	32.03	2264.26	8.77	480.95
XGBoost	89.60	14.24	31.51	2293.07	11.63	488.01
MLP	89.68	16.83	27.86	2038.53	7.79	436.14
Stateless LSTM	106.90	10.99	28.76	2438.63	7.73	518.60
Stateful LSTM	93.39	18.58	30.29	2606.34	8.74	551.47
Stacked Stateless LSTM	105.30	**9.69**	31.57	2729.41	8.07	576.81
Stacked Stateful LSTM	96.00	17.34	35.24	2372.24	8.15	505.79
CoT-Net	**80.16**	15.38	**24.86**	**1852.39**	**7.54**	**396.07**
AVERAGE	90.58	23.58	31.21	2195.51	8.67	

TABLE II: Classification Results (Mean Accuracy)

	AOL	BofA	Facebook	PayPal	Twitter	AVERAGE
ARIMA	0.44	0.39	0.31	0.41	0.43	0.40
Prophet	0.56	0.48	0.51	0.47	0.47	0.50
Logistic Regression	0.60	**0.70**	0.56	0.57	0.64	0.61
SVM-Poly	0.55	0.54	0.55	0.57	0.41	0.52
SVM-RBF	0.61	0.54	0.59	0.49	0.63	0.57
Random Forest	0.63	0.62	0.61	0.50	0.58	0.59
AdaBoost	0.53	0.66	0.60	0.56	0.61	0.59
MLP	0.59	0.60	0.56	0.51	0.62	0.58
Stateless LSTM	0.51	0.55	0.55	0.50	0.43	0.51
Stateful LSTM	0.62	0.64	0.62	0.56	0.53	0.59
Stacked Stateless LSTM	0.51	0.62	0.59	0.52	0.49	0.55
Stacked Stateful LSTM	**0.65**	0.66	0.59	0.50	0.53	0.59
CoT-Net	0.62	0.64	**0.65**	**0.62**	**0.66**	**0.64**
AVERAGE	0.57	0.59	0.56	0.52	0.54	

VI. Conclusions

In this study we introduce the idea of phishcasting — time series forecasting of phishing attack volume for a particular target organization. We develop a framework to leverage multiple (correlated) time series, additional structured data and alternate representations or embeddings for accurate forecasting, especially using complex time series data that lacks neat properties such as stationarity, seasonality and cyclicality. As part of our framework, we propose CoT-Net, a CNN-LSTM based deep neural network that learns time series embeddings for enhanced forecasting using complex, correlated time series data. Evaluation on 10 years of phishing data related to 5 target organizations' websites shows that traditional forecasting and standard machine learning methods are not well-suited for accurately predicting attack volumes. Our work is a first step towards the phishcasting idea — we hope that future work can further build on these concepts to develop models that attain better results in the same vein that other time series modeling problems have progressed over time.

References

[1] "Phishing activity trends reports." [Online]. Available: https://apwg.org/trendsreports/

[2] "Twitter says spear-phishing attack on employees led to breach," Jul 2020. [Online]. Available: https://www.theguardian.com

[3] B. Schneier, "Inside risks: semantic network attacks," *Communications of the ACM*, vol. 43, no. 12, p. 168, Jan 2000.

[4] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," in *Proceedings of the 2005 symposium on Usable privacy and security*. ACM, 2005, pp. 77–88.

[5] A. Abbasi, S. Sarker, and R. H. Chiang, "Big data research in information systems: Toward an inclusive research agenda," *Journal of the AIS*, vol. 17, no. 2, p. 3, 2016.

[6] R. T. Wright, M. L. Jensen, J. B. Thatcher, M. Dinger, and K. Marett, "Influence techniques in phishing attacks: An examination of vulnerability and resistance," *Info. Sys. Res.*, vol. 25, no. 2, pp. 385–400, 2014.

[7] A. Abbasi, F. M. Zahedi, and Y. Chen, "Phishing susceptibility: The good, the bad, and the ugly," in *2016 IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2016, pp. 169–174.

[8] F. M. Zahedi, A. Abbasi, and Y. Chen, "Fake-website detection tools: Identifying elements that promote individuals' use and enhance their performance," *Journal of the AIS*, vol. 16, no. 6, p. 2, 2015.

[9] Y. Chen, F. M. Zahedi, and A. Abbasi, "Interface design elements for anti-phishing systems," in *International Conference on Design Science Research in Information Systems*. Springer, 2011, pp. 253–265.

[10] A. Abbasi, D. Dobolyi, A. Vance, and F. M. Zahedi, "The phishing funnel model: A design artifact to predict user susceptibility to phishing attacks," *Information Systems Research*, pp. 1–25, 2021.

[11] H. Zhang, G. Liu, T. W. Chow, and W. Liu, "Textual and visual content-based anti-phishing: a bayesian approach," *IEEE transactions on neural networks*, vol. 22, no. 10, pp. 1532–1546, 2011.

[12] A. Abbasi, F. M. Zahedi, D. Zeng, Y. Chen, H. Chen, and J. F. Nunamaker Jr, "Enhancing predictive analytics for anti-phishing by exploiting website genre information," *Journal of MIS*, vol. 31, no. 4, pp. 109–157, 2015.

[13] M. L. Jensen, M. Dinger, R. T. Wright, and J. B. Thatcher, "Training to mitigate phishing attacks using mindfulness techniques," *Journal of Management Information Systems*, vol. 34, no. 2, pp. 597–626, 2017.

[14] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning : A systematic literature review: 2005-2019," 2019.

[15] A. Okutan, G. Werner, S. J. Yang, and K. McConky, "Forecasting cyberattacks with incomplete, imbalanced, and insignificant data," *Cybersecurity*, vol. 1, no. 1, pp. 1–15, Dec 2018.

[16] Z. Wang and Y. Zhang, "Ddos event forecasting using twitter data," in *Proc. of the IJCAI*, 2017, pp. 4151–4157.

[17] I. Perry, L. Li, and A. Okutan, "Differentiating and predicting cyberattack behaviors using lstm," in *IEEE Conf. on Dependable and Secure Computing*, 2018, pp. 1–8.

[18] G. Werner, S. Yang, and K. McConky, "Leveraging intra-day temporal variations to predict daily cyberattack activity," in *IEEE Intl. Conf. Intelligence and Security Informatics*, 2018, pp. 58–63.

[19] P. Filonov, A. Lavrentyev, and A. Vorontsov, "Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model," *NIPS Time Series Workshop*, 12 2016.

[20] B. Lim and S. Zohren, "Time series forecasting with deep learning: A survey," 2020.

[21] S. Ho, M. Xie, and T. Goh, "A comparative study of neural network and box-jenkins arima modeling in time series prediction," *Computers & Industrial Engineering*, vol. 42, no. 2-4, p. 371–375, 2002.

[22] A. Gasparin, S. Lukovic, and C. Alippi, "Deep learning for time series forecasting: The electric load case," 2019.

[23] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," 2020.

[24] M. Tadayon and Y. Iwashita, "Comprehensive analysis of time series forecasting using neural networks," 2020.

[25] D. G. Dobolyi and A. Abbasi, "Phishmonger: A free and open source public archive of real-world phishing websites," in *IEEE Conf. on Intelligence and Security Informatics*, 2016, pp. 31–36.