KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Technology

First Cycle 15 credits

Predicting the Options Expiration Effect Using Machine Learning Models Trained With Gamma Exposure Data

ALEXANDER DUBOIS

Stockholm, Sweden 2022

Major: Technology

Date: June 15, 2022

Supervisors:

Jonas Beskow, Professor at KTH Royal Institute of Technology

Jonas Thulin, Head of Asset Management at Erik Penser Bank

Host company: Erik Penser Bank

Examiner: Joakim Gustafsson

Swedish titel:

Prediktion av inverkan på aktiemarknaden då optioner upphör med hjälp av maskininlärningsmodeller tränade med dagliga GEX värden

Swedish abstract:

Flera studier har visat att optionsmarknaden påverkar aktiemarknaden, speciellt vid optioners utgångsdatum. Dock har få studier undersökt maskininlärningsmodellers förmåga att förutse denna effekt. I den här studien, implementeras och utvärderas fyra olika maskininlärningsmodeller, SVM, random forest, AdaBoost, och LSTM, med syftet att förutse om den underliggande aktiemarknaden stiger vid optioners utgångsdatum. Att optionsmarknaden påverkar aktiemarknaden vid optioners utgångsdatum beror på att market makers ombalanserar sina portföljer för att bibehålla en delta-neutral portfölj. Market makers behov av att ombalansera sina portföljer beror på åtminstone två variabler; gamma och antalet aktiva optionskontrakt. Därmed använder maskininlärningsmodellerna i denna studie GEX, som är en kombination av gamma och antalet aktiva optionskontrakt, med syftet att förutse om marknaden stiger vid optioners utgångsdatum. Vidare implementeras och utvärderas fyra olika varianter av LSTM modeller. Studien visar att en many-to-one LSTM modell med tre lager uppnådde bäst resultat med ett F1 score på 62%. Dock uppnådde ingen av modellerna bättre resultat än en modell som predicerar endast positiva klasser. Avslutningsvis diskuteras problematiken med att använda GEX och rekommendationer för framtida studier ges.

Predicting the Options Expiration Effect Using Machine Learning Models Trained With Gamma Exposure Data

Alexander J. G. Dubois

***Abstract*—The option expiration effect is a well-studied phenome, however, few studies have implemented machine learning models to predict the effect on the underlying stock market due to options expiration. In this paper four machine learning models, SVM, random forest, AdaBoost, and LSTM, are evaluated on their ability to predict whether the underlying index rises or not on the day of option expiration. The options expiration effect is mainly driven by portfolio rebalancing made by market makers who aim to maintain delta-neutral portfolios. Whether or not market makers need to rebalance their portfolios depend on at least two variables; gamma and open interest. Hence, the machine learning models in this study use gamma exposure (i.e. a combination of gamma and open interest) to predict the options expiration effect. Furthermore, four architectures of LSTM are implemented and evaluated. The study shows that a three-layered many-to-one LSTM model achieves superior results with an F1 score of 62%. However, none of the models achieved better predictions than a model that predicts only positive classes. Some of the problems regarding gamma exposure are discussed and possible improvements for future studies are given.**

***Impact Statement*—It is increasingly difficult for investors to gain an edge in today's stock market consisting of sophisticated players such as high-frequency and quantitative traders. This study evaluates four machine learning models, known to be suitable models for predicting stock market returns, with a previously unused input variable; daily gamma exposure. The result of this study gives both investors and academics a better understanding of machine learning models' ability to predict the options expiration effect using gamma exposure.**

***Index Terms*—AdaBoost, LSTM, Machine learning, Random forests, Stock markets, SVM**

I. Introduction

OPTIONS have been shown to impact the underlying stock market, especially on options' expiration dates [1][2]. This effect on the underlying stock market, close to options' expiration dates, is called the options expiration effect. One major driver behind this effect is the rebalancing made by market makers who aim to maintain delta-neutral portfolios, which is called delta-hedge rebalancing [1]. Delta-hedge rebalancing aims to hedge portfolios against changes in the options' prices once the underlying assets' prices change [3], which is often used by market makers as they try to profit on bid-ask spreads and not on directional bets [4]. At least two variables, gamma and open interest, have been shown to determine whether or not market makers need to rebalance their portfolios close to options' expiration dates [1][3]. Hence, this study uses a combination of gamma and open interest to predict whether the underlying stock index increase in value or not on options' expiration dates. Four machine learning models are trained and evaluated for this purpose.

This work was supported in part by Erik Penser Bank.

Alexander J. G. Dubois is a student at the KTH Royal Institute of Technology, Brinellvägen 8, Stockholm 11428 Sweden (e-mail: adub@kth.se)

A. The Problem Statement

No previous study, to the knowledge of the author, has used machine learning models with gamma and open interest as inputs to predict the options expiration effect. The goal of this study is hence to evaluate how well machine learning models can predict the options expiration effect using gamma and open interest as inputs. The research question of this study is hence:

How well can machine learning models predict the options expiration effect using gamma and open interest as inputs?

Apart from expanding the academic literature, the result of this study is of interest to the investor and finance community at large. As this study uses well-known machine learning models and publicly available information, the result of this study may provide all investors, regardless of information edges or resources, with a viable investment strategy. On a societal level, this study may lead to wealth creation.

A delimitation of this study is that it will only evaluate four machine learning models; SVM, random forest, AdaBoost, and LSTM. Furthermore, this study will only use daily GEX (i.e. a combination of gamma and open interest) values as input variables to the machine learning models. In addition, this study will only evaluate the options expiration effect on the S&P 500 index using SPX options (i.e. options contracts with the S&P 500 index as the underlying asset).

B. Previous Research

One study [5] showed that an SVM model is superior at predicting stock market returns compared to linear regression models. Another study [6] showed that normalization techniques improve the accuracy of SVMs for stock market predictions. The same study concluded that the choice of normalization method (e.g. min-max normalization, decimal scaling normalization) did not significantly affect the accuracy of the SVM. Moreover, for binary classification, a radial basis kernel function (RBF) has been shown to achieve superior

accuracy compared to linear, polynomial, and sigmoid kernels [7].

Another study [8] found that a random forest model achieves superior results (80,8% accuracy) compared to a deep learning model when trained on predicting stock market trends based on technical indicators. Furthermore, another study [9] showed that mean range normalization (see eq. 3) achieved the highest successful classification rate for random forest. In addition, another study [10] showed that AdaBoost predicted better results on periods greater than five days compared to the other tree-based models. Moreover, random forest models have been shown to handle noisy data better than AdaBoost [11]. For AdaBoost, no empirical study was found by the author evaluating the effect of normalization, which may be explained by the fact that AdaBoost utilizes normalization constants [12].

LSTM models have been shown to achieve better results compared to models such as random forest, AdaBoost, Gradient Boosting, and XGBoost at predicting stock market trends [10]. A previous comparative study [13] of activation functions and optimizers for LSTM networks, trained to predict stock market prices, found that a tanh activation function and an adam optimizer (i.e. a first-order gradient-based optimization algorithm [14]) achieved the leading accuracy. Furthermore, an LSTM model with a many-to-many architecture achieved superior accuracy in predicting the direction and timing of mid-price changes of stocks compared to a many-to-one architecture [15]. The same study proposed using three layers of LSTM (i.e adding two LSTM layers with a many-to-many architecture before the third many-to-one or many-to-many LSTM layer). In addition, another study [16] found that RNN-based models using tanh-estimator normalization (see eq. 4) achieve the lowest mean squared error and mean absolute error for stock market predictions compared to other normalization techniques. Furthermore, pre zero padding (i.e. the sequence is padded from the beginning) has been shown to give higher accuracy for LSTM networks compared to post zero padding [17].

A previous study [18] modeled stock prices with a Brownian Bridge and a Monte Carlo simulational to conclude that delta-hedge rebalancing by market makers impacts the options expiration effect. Furthermore, according to the same study, the effect of market makers' delta-hedge rebalancing is largest when market makers have long-gamma positions with relatively large open interest (i.e. the total amount of contracts that have not been executed or expired). One study [1] showed that there is a statistically significant relationship between gamma and the absolute return of the underlying stock. Furthermore, the same study concluded that a market maker's open interest determines the volume of trades needed to rebalance [1]. Hence, market makers' delta-hedge rebalancing depends on at least two variables; gamma and open interest. One study [3] proposes to multiply gamma by open interest for each strike price to include the effect of both gamma and open interest on market makers' decision whether to rebalance their portfolios or not, resulting in what the authors termed gamma exposure (GEX) [3]. Moreover, according to another study [10] stock market data is not independent as data from previous days affect the current data. This suggests that daily GEX values may not be independent, as both gamma and open interest depend on the underlying stock market.

To conclude, previous studies have shown that SVM, random forest, AdaBoost, and LSTM models are suitable models for predicting stock market returns. Furthermore, previous studies have suggested different preprocessing techniques and architectures for these models. In addition, gamma and open interest have been shown to impact market makers' delta-hedge rebalancing practices, which in turn impact the options expiration effect.

II. Theory

A. Gamma Exposure (GEX)

Delta is the derivative of an option's price with respect to the underlying asset's price. A delta-neutral portfolio is a portfolio where the sum of all positive and negative delta positions equals zero. Furthermore, gamma is the derivative of delta with respect to the underlying asset's price. To maintain delta-neutral portfolios, market makers hedge their positions based on gamma values [3]. For example, a larger gamma implies that delta will change more once the underlying asset's price changes. If a portfolio is gamma-neutral, changes in the underlying asset's price will not affect the portfolio's delta.

For a given options chain (see Tab. 1 for an example) gamma exposure (i.e. GEX) is calculated using eq. 1 [3]. That is, to multiply gamma (Γ) and open interest (OI) for each strike price (K), and multiply by negative one if it is a put option [3]. Negative one is multiplied for put options under the assumption that put options are sold by market makers which hence results in short gamma positions [3]. Moreover, the sum is multiplied by a hundred as a single option contract reflects a hundred shares of the underlying asset [3].

TABLE I
EXAMPLE OF THE STRUCTURE OF A DAILY OPTIONS CHAIN

Strike price (K)	Options type	Open interest (OI)	Gamma (Γ)
K_1	Call	OI_1	Γ_1
K_2	Call	OI_2	Γ_2
...	...	...	...
K_n	Put	OI_n	Γ_n

$$GEX = 100 \sum_{k=1}^{n} OI_k \Gamma_k \begin{cases} -1, & \text{if put option} \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

B. Support Vector Machine (SVM)

Support vector machines (SVMs) are machine learning models that aim to separate categorical data points in space by one or more hyperplanes [5]. A standard SVM is trained to maximize the margin between two given categories [5]. SVM classifiers can use different kernels (i.e. separating functions) such as a polynomial or linear function [19]. Compared to neural networks that may find local optimum solutions, a standard SVM optimizes a convex function [20] and hence finds a global optimum [6]. This makes SVM less prone to overfitting compared to neural networks [6].

C. Random Forest

Random forest is an ensemble model that combines multiple tree predictors to classify categories [21]. The input to each tree contains randomly independently selected features with the same distribution [21]. By selecting the features at random the model can decrease the correlation between the trees resulting in more accurate predictions [22], and a lower generalization error [21] (see eq. 2). In addition, the model also selects the training data at random [10]. For categorical predictions, the model calculates the final prediction as the majority vote of the results from each individual tree [23].

A random forest model's generalization error (PE) has an upper bound (see eq. 2) which depends on the strength, defined as the expected value of the margin function, of the individual trees (s) and the correlation (c) between the trees' individual raw margin functions [21].

$$PE \leq \frac{c(1-s^2)}{s^2} \tag{2}$$

Furthermore, the Strong Law of Large Numbers proves that random forest models do not overfit with an increasing number of trees [21], and hence overcome the overfitting problem associated with decision tree algorithms [22]. In addition, random forest models are also less sensitive to outlier data compared to decision tree algorithms [22].

D. AdaBoost

AdaBoost is a boosting algorithm[12]. Boosting is a method to reduce the error of weak (i.e. slightly better than random guessing) learning algorithms [12]. The method combines the result from repeatedly running the weak learning algorithm on samples of the training data [12]. On each iteration, the weak learner gives a hypothesis [24]. During training, AdaBoost aims to find a hypothesis with a low weighted error relative to the current distribution [24]. The final hypothesis of the AdaBoost model is computed as the sign of the weighted hypothesis. In addition, AdaBoost uses a normalization constant in each iteration so that the next distribution is a probability density function with an aggregate probability of one [12].

E. Long Short Term Memory (LSTM)

Feedforward neural networks assume that two successive inputs are independent [10]. In contrast, recurrent neural networks (RNNs) recurrently use information from previous iterations to accommodate for the dependence between sequential inputs [10]. However, standard RNNs have a vanishing gradient problem (i.e. the sensitivity to earlier inputs decays as the hidden layer is updated by newer inputs) [25]. LSTM networks are similar to standard RNNs except that LSTM models use memory blocks instead of summation units in the hidden layer [25]. These memory blocks allow LSTM networks to preserve gradient information over longer periods which solves the vanishing gradient problem [25]. This makes LSTM networks more suitable compared to standard RNNs for problems requiring longer intervals of contextual data [25]. Each memory block contains a forget gate, input gate, and output gate. The forget gate controls which information to disregard, the input gate controls the amount of new information to remember, and the output gate controls the amount of information to be outputted into the new iteration [10]. Each gate contains weights and biases that are updated during the training process [10]. An LSTM network is usually trained using gradient descent with a forward and backward pass [10]. Furthermore, there are different LSTM architectures (e.g. many-to-many, many-to-one) [15]. A many-to-many architecture outputs the hidden states at each timestep while a many-to-one architecture only outputs the hidden state at the final layer [15].

F. Normalization Techniques

Mean range normalization is equivalent to min-max normalization onto the range 0 to 1 (see eq. 3).

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3}$$

Tanh-estimator normalization yields values in the 0 to 1 range (see eq. 4).

$$x_{\text{norm}} = 0.5\left(\tanh\left(\frac{0.01(x-\mu)}{\delta}\right) + 1\right) \tag{4}$$

Where δ and μ are standard deviation and mean value of the input data.

III. Method

A. Data

The input data consists of daily gamma exposure (GEX) for SPX options (i.e. options contracts with the S&P 500 index as the underlying asset) grouped by options expiration date. For each option expiration date, the output variable is one if the S&P 500 index closing price was greater than the previous day's closing price and zero otherwise. The daily GEX value and daily closing price of the S&P 500 index starting from the 6th of May 2011 until the 26th of April 2022 were collected publicly from SqueezeMetrics Research [26]. The historic options' expiration dates were scraped from MarketWatch using the BeautifulSoup4 library with Python 3.7. As the input and output data pairs are grouped by options expiration date, which occurs once a month for standard options contracts, the number of data points in the dataset is 132. The number of days between each option's expiration date varies between 15 and 25 days, depending on the number of days in each month and bank holidays. Hence the number of GEX values per input varies. This study used zero padding to make the input size fixed.

TABLE II
EXAMPLE OF OUTPUT AND INPUT DATA WITHOUT ANY NORMALIZATION TECHNIQUE APPLIED.

GEX_1	GEX_2	...	GEX_{25}	y
0	24 218 668 213	...	-6 498 722 661	1
..	...	...	...	...
0	0	...	10 318 466 718	0

The daily GEX values range from -7 496 822 661 to 24 218 668 213. A tanh estimator normalization technique was

implemented for LSTM using Numpy and eq. 4, resulting in a 0 to 1 range for the daily GEX values. A mean range normalization was used for both SVM and random forest, which was implemented using scikit-learn's MinMaxScalar function with the default range of 0 to 1. No normalization technique was used for AdaBoost. To evaluate the normalization techniques SVM, random forest, and LSTM were also trained and evaluated using no normalization technique.

B. Machine Learning Models

SVM, random forest, and AdaBoost were implemented using SVC, RandomForestClassifier, and AdaBoostClassifier from scikit-learn respectively. For SVC a radial basis kernel function (RBF) was used. For LSTM four different architectures were implemented; single LSTM many-to-one, single LSTM many-to-many, three-layered LSTM many-to-one, and three-layered LSTM many-to-many (see Fig. 1).

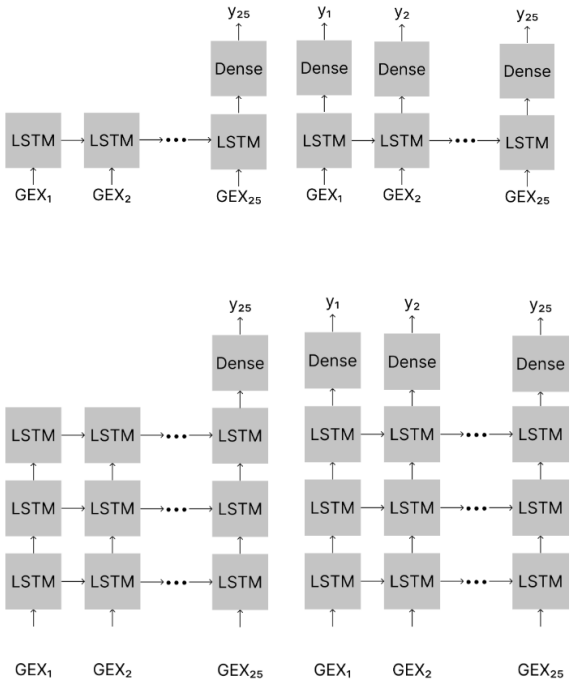

Fig. 1. Illustration of LSTM models. Single LSTM many-to-one (top left), single LSTM many-to-many (top right), three-layered LSTM many-to-one (bottom left), and three-layered LSTM many-to-many architecture (bottom right).

All the four architectures were implemented using Kera's Sequential model with LSTM layer(s) followed by a dense layer. As performed in a previous study [15], each of the four LSTM architectures was implemented and evaluated with 20, 30, and 50 LSTM units. Each LSTM layer used an input shape of (25,1) as the maximum days before expiration is 25 days. The dense layer used one unit and a sigmoid activation function, which is suitable for the binary classification problem in this study [27]. Furthermore, each sequential Keras model used an adam optimizer and binary cross-entropy loss function. For training, 124 epochs and a batch size of 64 were used.

C. Evaluation

The dataset was divided into training and test datasets containing 80% and 20% of all datapoints respectively. All datapoints were divided into the two datastes sequentially starting with the training dataset followed by the test dataset. The training set was used for training the models and the test set was used to evaluate the models.

F1 score, precision, and recall were used as evaluation metrics. Precision is the proportion of positive predictions that were correctly classified (see eq. 5). Recall is the proportion of positives that were predicted correctly (see eq. 6). F1 score is the harmonic mean of precision and recall (see eq. 7).

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 - score = \frac{TP}{TP + 0,5(FP + FN)} \tag{7}$$

Where TP, FP, TN, FN are true positives, false positives, true negatives, and false negatives respectively.

IV. Result

TABLE III

Model	Normalization technique	F1-score	Precision	Recall
SVM	None	0,25	0,29	0,22
SVM	Mean range	**0,48**	0,35	**0,78**
Random Forest	None	0,30	0,27	0,33
Random Forest	Mean range	0,40	**0,36**	0,44
AdaBoost	None	0,20	0,18	0,22

TABLE IV

LSTM architecture	Normalization technique	Units	F1-score	Precision	Recall
many-to-one	None	20	0,52	0,39	0,78
		30	0,22	0,22	0,22
		50	0,38	0,33	0,44
	tanh estimator	20	0,51	0,35	1,00
		30	0,51	0,39	0,78
		50	0,51	0,39	0,78
many-to-many	None	20	0,56	0,39	1,0
		30	0,56	0,39	1,0
		50	0,56	0,39	1,0
	tanh estimator	20	0,51	0,35	1,0
		30	0,51	0,35	1,0
		50	0,51	0,35	1,0
three-layered many-to-one	None	20	0,48	0,38	0,67
		30	0,50	0,37	0,78
		50	**0,62**	**0,45**	**1,00**
	tanh estimator	20	0,52	0,39	0,78
		30	0,52	0,39	0,78
		50	0,56	0,39	1,00
three-layered many-to-many	None	20	0,55	0,38	1,00
		30	0,56	0,39	1,00
		50	0,52	0,36	0,89
	tanh estimator	20	0,51	0,35	1,00
		30	0,51	0,35	1,00
		50	0,51	0,35	1,00

V. Discussion

Based on the results the three-layered many-to-one LSTM model, using no normalization technique and with 50 units, achieved superior results with a F1 score of 62%. The same model also achieved the highest precision of 45% among the models. However, the model also has a recall of 100% which means that the model predicted zero false negatives. As the precision is less than 50% the number of false positives is greater than the number of true positives. With no false negatives and a greater number of false positives than true positives, this model is likely trained to overly predict positive classes. Moreover, as the data consists of a ∼50% proportion between positive and negative classes and as all the models (except SVM with no normalization) achieved higher recall than precision (i.e. more false positives than false negatives) all of the models tended to overly predict positive categories. Furthermore, with a ∼50% split between positive and negative classes, a model that only predicts positive classes would achieve an F1 score of ∼66% (see eq. 7). Hence, none of the models in this study achieves better results than a model that only predicts positive classes. This result is consistent with the efficient market hypothesis that states that publicly available information, such as GEX, is fully reflected in asset prices and hence it is impossible to achieve excess return based on GEX [28].

Even though this study's result is consistent with the efficient market hypothesis, it does not necessarily mean that one can not predict the options expiration effect using gamma and open interest with machine learning models. First of all, some previous studies have found that the market is not fully efficient [29]. Moreover, one possible reason that all of the models achieved inferior results compared to a model that solely predicts positive classes may be that GEX does not encapsulate market makers' need to rebalance their portfolios correctly. For example, GEX assumes that all put options are sold by market makers, which is not the case in practice [30]. By using open interest (i.e. all active contracts), GEX also assumes that all market participants hedge their deltas [3], which most investors using a directional strategy do not. Another problematic assumption of GEX is the assumption that market makers trade to maintain an exact portfolio delta regardless of transactional costs [3], which likely is unrealistic. Also, GEX is calculated by aggregating the sum of gamma and open interest for each strike price (see eq. 1) which may lead to a loss of information. Hence GEX may be too inaccurate or too simplistic to capture market makers' delta-hedge rebalancing practices.

Nevertheless, based on the results one can also conclude that using a mean range normalization technique for the GEX input variables for SVM and random forest is viable as it increased their precision, recall, and F1 score. Furthermore, AdaBoost achieved the lowest precision, recall, and F1 score of all models which suggests that it is the least suitable model of the four for predicting the options expiration effect based on GEX. One possible reason that random forest achieved better results than AdaBoost is that GEX data may be noisy which random forest models can handle better [11].

A. Future Work

Vanna (i.e. the derivative of delta with respect to the implied volatility of the underlying asset), moneyness, implied volatility, and order type (i.e. short or long position) have been shown to impact market makers' need to rebalance their portfolio [30]. Moreover, dealer directional open interest (DDOI) is a more accurate approximation of market makers' open interest [30]. Hence the result could likely be improved by including these variables as inputs to the machine learning models. To include all the suggested variables above, one could use an options chain as input containing these variables separated for each strike price. This solution would mitigate the risk of information loss due to aggregation. Furthermore, this solution would also be able to accommodate for moneyness, as it depends on the difference between the option's strike price and the underlying asset's price. It would also increase the number of input variables significantly which may help the LSTM model learn not to overpredict positive classes. A recommendation for future studies is hence to include and evaluate the LSTM model using these variables, for each strike price, as inputs.

Another possible improvement for future studies is to evaluate more machine learning models. For example, a bi-directional LSTM model has been shown to achieve superior results compared to LSTM models at predicting stock market trends [31]. The same study also concluded that the performance of LSTM and bi-directional LSTM models vary widely depending on the choice of hyperparameters (e.g. number of epochs, number of LSTM units). Future studies are hence also suggested to use a hyperparameter tuning process and a bi-directional LSTM model.

Furthermore, an RNN-based model (i.e. LSTM) was used in this study as previous studies suggested that sequential stock market price changes are not independent of each other [10], which made RNN a suitable choice for this study. In contrast, random walk theories suggest that sequential stock market price changes are independent [29]. This may suggest, as both gamma and open interest depends on stock price changes, that daily GEX values are independent. To examine this varying opinion of independency, future studies are suggested to compare the performance of a feed-forward neural network compared to an RNN-based model (e.g. LSTM).

B. Ethical Perspective

Most of the variables suggested for future studies are not publicly available for each strike price. They are only available for a large sum of money. Hence the research result from such a study may only help well-funded investors gain an edge over less-funded investors, which would be problematic from an ethical perspective. On a societal level, it could possibly widen the wealth gap. Furthermore, the vastly larger input sizes (∼1000 times larger) would likely require more electricity for training, which may come from unsustainable sources, such as coal-powered energy plants, which would harm the environment.

VI. Conclusion

A three-layered many-to-one LSTM model, using no normalization technique and with 50 LSTM units, achieved the top results (62% F1 score) when predicting the options expiration effects based on daily GEX data. However, all models performed worse than a model that solely predicts positive classes. The result of this study is valuable for the investor and finance community as it suggests that one cannot achieve excess returns using solely daily GEX values as input values to SVM, random forest, AdaBoost, and LSTM models. Furthermore, recommendations for how future research may achieve better results are given, such as using more variables and a bi-directional LSTM model.

References

[1] S. X. Ni, N. D. Pearson, A. M. Poteshman, and J. S. White, "Does option trading have a pervasive impact on underlying stock prices?" *The Review of Financial Studies*, vol. 34, no. 4, pp. 1952–1986, Apr. 2021.

[2] E. M. Cinar and J. Vu, "Evidence on the effect of option expirations on stock prices," *Financial Analysts Journal*, vol. 43, no. 1, pp. 55–57, Jan. 1987.

[3] "Gamma exposure (gex)," SqueezeMetrics Research, New York City, NY, USA, Tech. Rep., Dec. 2017. [Online]. Available: https://squeezemetrics.com/monitor/download/pdf/white_paper.pdf

[4] S. W. Poser, "Market makers in financial markets: Their role, how they function, why they are important, and the nyse dmm difference," NYSE Group, Inc, New York City, NY, USA, Tech. Rep., Sep. 2021. [Online]. Available: https://www.nyse.com/publicdocs/nyse/NYSE_Paper_on_Market_Making_Sept_2021.pdf

[5] V. Gururaj, V. R. Shriya, and K. Ashwini, "Stock market prediction using linear regression and support vector machines," *International Journal of Applied Engineering Research*, vol. 14, no. 8, pp. 1931–1934, Aug. 2019.

[6] J. Pan, Y. Zhuang, and S. Fong, "The impact of data normalization on stock market prediction: Using SVM and technical indicators," in *Second International Conference on Soft Computing in Data Science*, Kuala Lumpur, Malaysia, 2016, pp. 72-88.

[7] Y. Bao, T. Wang, and G. Qiu, "Research on applicability of SVM kernel functions used in binary classification," in *Proceedings of International Conference on Computer Science and Information Technology*, S. Patnaik and X. Li, Eds., New Delhi, India, 2014, pp. 833-844.

[8] R. Cervelló-Royo and F. Guijarro, "Forecasting stock market trend: A comparison of machine learning algorithms," *Finance, Markets and Valuation*, vol. 6, no. 1, p. 37–49, Jan. 2020.

[9] Z. Ihsan, M. Y. Idris, and A. H. Abdullah, "Attribute normalization techniques and performance of intrusion classifiers: A comparative analysis," *Life Science Journal*, vol. 10, no. 4, pp. 2568–2576, Dec. 2013.

[10] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. Shahab, "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, July 2020.

[11] Y. Chang, W. Li, and Z. Yang, "Network intrusion detection based on random forest and support vector machine," in *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, Guangzhou, China, 2017, pp. 635-638.

[12] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning*, Bari, Italy, 1996, pp. 148–156.

[13] M. Rana, M. Uddin, and M. Hoque, "Effects of activation functions and optimizers on stock price prediction using LSTM recurrent networks," in *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, New York, NY, USA, 2019, pp. 354-358.

[14] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. Presented at ICLR 2015. [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf

[15] Y. Tu, "Predicting high-frequency stock market by neural networks," M.S. thesis, Dept. Mathematics, Imperial College London, London, United Kingdom, 2020.

[16] S. Bhanja and A. Das, "Impact of data normalization on deep neural network for time series forecasting," unpublished.

[17] M. Dwarampudi and N. V. S. Reddy, "Effects of padding on lstms and cnns," unpublished. [Online]. Available: https://arxiv.org/pdf/1903.07288.pdf

[18] M. Avellaneda and M. D. Lipkin, "A market-induced mechanism for stock pinning," *Quantitative Finance*, vol. 3, no. 6, pp. 417–425, Aug. 2003.

[19] B. Schölkopf, S. Mika, C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1000–1017, Sep. 1999.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, Sep. 1995.

[21] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.

[22] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues*, vol. 9, no. 3, pp. 272–278, Sep. 2012.

[23] M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[24] R. E. Schapire, "Explaining AdaBoost," in *Empirical Inference*, B. Schölkopf, Z. Luo, and V. Vovk, Eds., Berlin, Germany: Springer, 2013, pp. 37-52. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-41136-6_5

[25] A. Graves, "Long short-term memory," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 37-44. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-3-642-24797-2.pdf

[26] SqueezeMetrics Research, June 2022. [Online]. Available: https://squeezemetrics.com/monitor/dix#gex

[27] A. V. Olgac and B. Karlik, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *International Journal of Artificial Intelligence And Expert Systems*, vol. 1, no. 4, pp. 111–122, Feb. 2011.

[28] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The Journal of Finance*, vol. 25, no. 2, pp. 383–417, May 1970.

[29] B. G. Malkiel, "The efficient market hypothesis and its critics," *Journal of Economic Perspectives*, vol. 17, no. 1, pp. 59–82, 2003.

[30] "The implied order book: Measuring s&p500 liquidity with spx options," SqueezeMetrics Research, New York City, NY, USA, Tech. Rep., July 6 2020. [Online]. Available: https://squeezemetrics.com/download/The_Implied_Order_Book.pdf

[31] A. I. Sunny, M. M. S. Maswood and A. G. Alharbi, "Deep learning-based stock price prediction using LSTM and bi-directional LSTM model," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, Cairo, Egypt, 2020, pp. 87-92.

Alexander J. G. Dubois is currently pursuing a B.S. in industrial engineering and management at KTH Royal Institute of Technology, Stockholm, Sweden.

During the summer of 2021, he worked as a Data Science Intern at Debricked AB. He also worked for five months ending April 2022 as a Junior Analyst within the asset management team at Erik Penser Bank.

TRITA-EECS-EX-2022:345