UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
ESCOLA DE INFORMÁTICA APLICADA

REDES NEURAIS RECORRENTES APLICADAS À PREDIÇÃO DE VALORES DE AÇÕES

BRENO PHILLIP MENDES BARONTE

Orientador
PEDRO NUNO DE SOUZA MOURA

RIO DE JANEIRO, RJ – BRASIL
JANEIRO DE 2022

Catalogação informatizada pelo autor

B265

BARONTE, BRENO PHILLIP MENDES
REDES NEURAIS RECORRENTES APLICADAS À PREDIÇÃO DE VALORES DE AÇÕES / BRENO PHILLIP MENDES BARONTE. -- Rio de Janeiro, 2022.
62

Orientador: PEDRO NUNO DE SOUZA MOURA.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal do Estado do Rio de Janeiro, Graduação em Sistemas de Informação, 2022.

1. deep learning. 2. redes neurais recorrentes. 3. predição. 4. ações. 5. bolsa de valores. I. NUNO DE SOUZA MOURA, PEDRO, orient. II. Título.

REDES NEURAIS RECORRENTES APLICADAS À PREDIÇÃO DE VALORES DE AÇÕES

BRENO PHILLIP MENDES BARONTE

Projeto de Graduação apresentado à Escola de Informática Aplicada da Universidade Federal do Estado do Rio de Janeiro (UNIRIO) para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado por:

_____

Prof. Pedro Nuno de Souza Moura, D.Sc. (UNIRIO)

_____

Profa. Geiza Maria Hamazaki da Silva, D.Sc. (UNIRIO)

_____

Profa. Morganna Carmem Diniz, D.Sc. (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

JANEIRO DE 2022

Agradecimentos

Primeiramente agradeço à minha família. Em especial ao meu pai, Edy Carlos Santos Mendes, que sempre lutou para que eu tivesse acesso a melhor educação possível dentro da nossa realidade e minha mãe, Monica Elisa Siqueira Gomez Beltran de Baronte, por seu amor. Ao meu irmão, Alan Baronte Pfeffer, e ao segundo pai que recebi, Alex Ventura Pfeffer. Também agradeço à minha avó, Maria Lucia, por seu suporte incondicional, e a todos os demais familiares.

Ao professor que me orientou neste projeto, Pedro Nuno de Souza Moura, por todo o seu esforço, atenciosidade, parceria e inspiração profissional e pessoal.

À banca, professoras Morganna Carmem Diniz e Geiza Hamazaki, por terem aceitado o convite, pelo suporte que recebi de ambas durante a graduação e pelo exemplo impecável como professoras e diretoras.

Aos meus amigos discentes Elias Henrique, Gabriel Trindade, Marcos José, Mariana Oliveira, Tomas Sepulveda e Victor Cezar por toda a camaradagem durante a graduação.

Aos professores Beatriz Malajovich e Gleison Santos que conseguiram tornar a graduação mais leve através de simples conversas, e a todos da UNIRIO que contribuíram.

Aos meus colegas de trabalho Alexandro Pinheiro, João Ricardo, Thomás Pereira e Victor Gonçalves por terem sido mais que colegas de trabalho.

RESUMO

O número de CPFs cadastrados na bolsa de valores brasileira apresentou um aumento surpreendente nos últimos anos e mostra que o brasileiro está mais disposto a entrar na esfera dos investimentos. É de conhecimento comum que os preços de uma ação comercializada em uma bolsa de valores possuem movimentos imprevisíveis. Tal comportamento é devido às muitas variáveis que estão envolvidas no seu processo de valorização e na maioria das vezes não são nítidas o suficiente para serem identificadas. Neste cenário com novos investidores inexperientes em paralelo aos avanços atuais da inteligência artificial, se faz interessante o estudo de como técnicas de aprendizagem profunda podem ser úteis na tarefa de predição de valores de ações. Este trabalho utiliza dados de 20 anos da ação de uma empresa norte-americana e apresenta 480 experimentos com redes neurais recorrentes. O desempenho alcançado por cada um dos experimentos é registrado, comparado com os demais e apresentam resultados interessantes.

Palavras-chave: *deep learning*, redes neurais recorrentes, predição, ações, bolsa de valores.

ABSTRACT

The number of registered CPFs on the Brazilian stock exchange has shown a surprising increase in recent years and shows that Brazilians are more willing to enter the investment sphere. It is common knowledge that the prices of a stock traded on a stock exchange have unpredictable movements. Such behavior is due to the many variables that are involved in its valuation process and most of the time they are not clear enough to be identified. In this scenario with new inexperienced investors in parallel with current advances in artificial intelligence, it is interesting to study how deep learning techniques can be useful in the task of predicting stock values. This work uses data from 20 years of the stock of a North American company and presents 480 experiments with recurrent neural networks. The performance achieved by each one of the experiments are registered, compared with the others and present interesting results.

Keywords: deep learning, recurrent neural networks, prediction, stocks, stock exchange.

Índice

Introdução 12
Motivação 12
Objetivos 14
Organização do texto 15

Conceitos Preliminares 16
Conceitos iniciais 16
Breve história do Deep Learning 16
Algumas Arquiteturas de Redes Neurais 18
Rede Neural Direta (Feedforward) 18
Redes Neurais Recorrentes 20
Recurrent Neural Networks (RNN) 20
Exploding Gradient e Vanishing Gradient 21
Long Short-Term Memory (LSTM) 23
Gated Recurrent Unit (GRU) 27
Mercado de Ações 29

Revisão Bibliográfica 31

Modelos Adotados 34

Experimentos Computacionais 37
Dataset Utilizado 37
Ambiente Computacional 38
Códigos Utilizados e Detalhes dos Experimentos 38
Análise 41

Conclusão 45

Referências Bibliográficas 48

LISTA DE ABREVIATURAS E SIGLAS

BOVESPA - Bolsa de Valores de São Paulo

CPF - Cadastro de Pessoas Físicas

DAX - Deutscher Aktienindex

GRU - Gated Recurrent Unit

HSI - Hang Seng Index

LSTM - Long Short-Term Memory

MAE - Mean Absolute Error

MLP - Multi-Layer Perceptron

MSE - Mean Squared Error

NASDAQ - National Association of Securities Dealers Automated Quotations

RMSE - Root Mean Square Error

RNN - Recurrent Neural Networks

SGD - Stochastic gradient descent

SVM - Support Vector Machine

S&P - Standard and Poor's

XGBoost - Extreme Gradient Boosting

Índice de Apêndices

Apêndice 1 51

Índice de Tabelas

Tabela 1 - Partes do código identificador gerado pelo primeiro projeto. 40

Tabela 2 - Resultados absolutos dos experimentos. 41

Tabela 3 - Os cinco experimentos com os melhores resultados (menores erros). 42

Tabela 4 - Os cinco experimentos com os piores resultados (maiores erros). 42

Tabela 5 - Dados dos erros dos experimentos que utilizaram os datasets com todos os registros. 43

Tabela 6 - Dados dos erros dos experimentos que utilizaram os datasets com metade dos registros. 43

Tabela 7 - Resultados do número de vezes em que cada modelo de camadas apresentou melhor desempenho. 44

Índice de Figuras

Figura 1 - Inteligência Artificial x Aprendizagem de Máquina x Aprendizagem Profunda. 12

Figura 2 - Rede neural artificial (Feedforward). 18

Figura 3 - Rede neural artificial capaz de identificar um pão com ovo. 19

Figura 4 - Exemplo de funcionamento da neural artificial capaz de identificar um pão com ovo. 20

Figura 5 - Comparação macro da estrutura de uma rede *feedforward* e uma *RNN*. 21

Figura 6 - Exemplo simplificado de uma rede neural do tipo LSTM. 24

Figura 7 - Primeiro passo no processamento de uma LSTM. 25

Figura 8 - Segundo passo no processamento de uma LSTM. 26

Figura 9 - Terceiro passo no processamento de uma LSTM. 27

Figura 10 - Exemplo simplificado de uma rede neural do tipo GRU. 28

Figura 11 - Modelos usados que utilizam LSTM nos experimentos. 35

Figura 12 - Modelo usado que utiliza GRU nos experimentos. 36

Figura 13 - Exemplo de grupo na análise dos modelos de camadas. 44

1 Introdução

1.1 Motivação

"Aprendizagem profunda é um tipo específico de aprendizagem de máquina" (GOODFELLOW et al, 2016). A aprendizagem de máquina (*Machine Learning*) é um dos ramos da inteligência artificial. Por ora vamos dizer que a Aprendizagem Profunda (*Deep Learning*) é uma área que nos permite ensinar uma máquina a realizar determinadas tarefas a partir da tentativa de reprodução das redes neurais do cérebro humano. A Figura 1 mostra a relação de subconjunto entre as três áreas citadas.

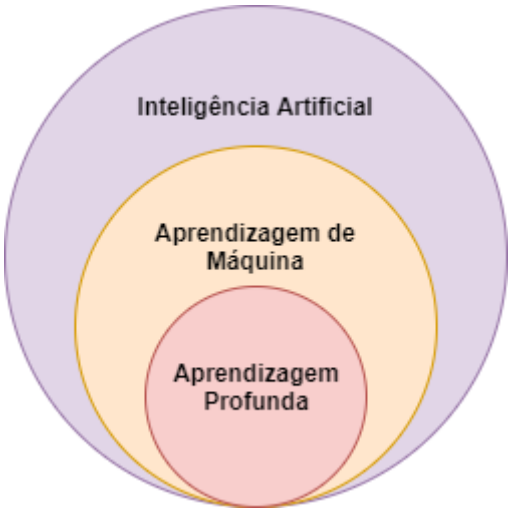

Figura 1 - Inteligência Artificial x Aprendizagem de Máquina x Aprendizagem Profunda.

Em 2012 ocorreu um grande marco no campo de visão de máquina, um dos tópicos mais estudados em aprendizagem profunda. Utilizando o recém-criado banco de imagens "ImageNet"[1], o time da universidade de Toronto conseguiu uma acurácia de 75% através da AlexNet, a rede neural desenvolvida para a competição (HOUSE, 2021). Uma diferença importante entre esta e as demais redes neurais da competição foi que a AlexNet foi a única que utilizava aprendizagem profunda e a que obteve o melhor resultado. Dessa forma, o campo de aprendizagem profunda ganhou um destaque na área da inteligência artificial.

Na atualidade, a aprendizagem profunda, visão de máquina e segmentos relacionados permeiam o nosso cotidiano de formas simples e sutis. O Instagram utiliza reconhecimento facial para a função de filtros, por exemplo. Outra área muito explorada é o reconhecimento de voz, utilizado pelas mais diversas marcas e seus produtos como a

[1] Disponível em: <https://www.image-net.org/>. Acesso em: 15 Nov. 2021.

Microsoft com a Cortana, Amazon com a Alexa, Apple com a Siri, Google com o Google Assistant, e outros mais.

O presente trabalho mostra que a aprendizagem profunda também pode ser utilizada em cenários em que a sua aplicação ainda não é amplamente divulgada, como é o caso para variações de preço em ações da Bolsa de Valores.

> A Bolsa de Valores tem origem remota, tendo registros tanto na história grega quanto na romana, como um local em que as pessoas se reuniam para trocar e negociar mercadorias. A palavra Bolsa, para esse local, tem origem na Bélgica, na cidade de Bruges, mais especificamente na casa de Van der Burse. Lá, aconteciam reuniões de comerciantes e mercadores e, na entrada da casa, havia o brasão da família, que possuía três bolsas desenhadas – essa relação ficou marcada e se expandiu. Anos depois, em 1141, Luís VII criou a Bourse de Paris, que foi regulamentada em 1304.
>
> No Brasil, a Bolsa de Valores de São Paulo, a BOVESPA (como era conhecida anteriormente) foi fundada em 23 de agosto de 1890 por um grupo de agentes de negócios liderado por Emílio Rangel Pestana – era a Bolsa Livre de São Paulo. Infelizmente, a empreitada não durou nem dois anos, devido à política econômica da época. No entanto, novamente, em 1895, Emílio prosseguiu com a ideia, abrindo a Bolsa de Fundos Públicos de São Paulo, que negociava principalmente títulos públicos – era o embrião da Bovespa. Essa ganhou esse nome apenas em 1934, ano em que saiu de onde estava e foi transferida para o Palácio do Café (INFOMONEY, 2021).

Atualmente, a bolsa de valores é o local onde são negociadas as ações (INFOMONEY, 2021). Uma ação é um produto de renda variável, ou seja, seu valor de compra e venda varia conforme o tempo passa e não é possível saber se esta irá valorizar ou desvalorizar o seu preço no momento em que é comprada (INFOMONEY 3, 2021). Vale salientar que sempre que alguém compra a ação de uma empresa, este se torna um sócio dela (INFOMONEY 2, 2021). Dessa maneira, este trabalho foca somente neste produto, as ações, já que a bolsa de valores possui muitos outros produtos e cada um deles tem suas particularidades.

A bolsa de valores brasileira teve o número de CPFs cadastrados na faixa de 500.000 até 600.000 no intervalo de tempo entre os anos 2008 e 2016, com exceção de 2010 onde o número superou 610.000. Todavia é possível ver que o número de pessoas interessadas na bolsa cresceu muito nos últimos anos, referente ao intervalo de 2017 até

2020 (posição do mês de novembro) com os respectivos números: 619.625, 813.291, 1.681.033, 3.173.411 (B3, 2021).

Devido ao grande número de novos investidores, é esperado que os iniciantes no meio de investimentos tenham muitas dúvidas e mais perguntas do que respostas. É o previsto devido à exposição a um novo meio. Não sabemos quando uma ação irá subir ou descer, quando devemos comprar ou vender, e outras dúvidas além destas.

Com a realidade supracitada e os recentes avanços de aprendizagem profunda, é oportuno que se crie um modelo que seja capaz de predizer movimentos de preços de uma ação. Esse modelo poderá ser um auxílio ao investidor iniciante, assim como para o mais experiente, na hora da tomada de decisão em relação à negociação de ações.

Todavia, é importante ressaltar que nenhum dos modelos desenvolvidos no presente trabalho possui o objetivo de ser assertivo em operações reais. Este não é o propósito deste trabalho. As bolsas de valores possuem movimentos ainda muito imprevisíveis e de difícil especulação. Os modelos gerados neste trabalho não se propõem a servir como um mecanismo para recomendação de investimentos e isso não é incentivado.

Uma motivação adicional da parte pessoal do autor é que este trabalho serviu como oportunidade de estudar e se aprofundar em assuntos que eram até então desconhecidos para o mesmo, utilizando este trabalho de forma a complementar sua formação acadêmica.

1.2 Objetivos

O objetivo deste trabalho foi estudar o problema de predição de valores de ações a partir de dados históricos sob o ponto de vista da Aprendizagem Profunda e propomos algumas arquiteturas de redes neurais para realizar essa tarefa. Tivemos como objetivo que essas arquiteturas propostas possuíssem índices de erro mínimo dentro do nosso contexto.

Mais especificamente, estudamos, desenvolvemos e testamos diversas arquiteturas de redes neurais com o propósito de prever o preço futuro de uma determinada ação da bolsa de valores. Diversos experimentos foram conduzidos de modo a validar os modelos e obter o seu grau de confiabilidade.

Desse modo, essa abordagem proposta pode servir como mais uma fonte de suporte à tomada de decisão para o investidor na hora de operar no mercado de ações. Além disso, todos os artefatos gerados e utilizados para a composição deste trabalho são disponibilizados a fim de facilitar possíveis trabalhos futuros e também para uma melhor

compreensão dos assuntos abordados. Isso inclui os códigos-fontes e os resultados dos experimentos em sua totalidade.

1.3 Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: Conceitos preliminares - Apresenta conceitos introdutórios que serviram como base do trabalho na área de Inteligência Artificial e conceitos relacionados ao mercado de ações.

- Capítulo III: Revisão bibliográfica - Revisão de trabalhos relacionados.

- Capítulo IV: Modelos adotados - Exibe os modelos que foram adotados para os experimentos computacionais.

- Capítulo V: Experimentos computacionais - Mostra o ambiente computacional usado, a base de dados, a metodologia e os resultados dos experimentos.

- Capítulo VI: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

2 Conceitos Preliminares

2.1 Conceitos iniciais

Este trabalho já mencionou que *Machine Learning* é um dos ramos da Inteligência Artificial e que *Deep Learning* é um sub-ramo desse, porém o capítulo atual visa a elucidar alguns conceitos que já foram citados na introdução e ainda não aprofundados.

Segundo Andrew Moore, ex-reitor da escola de Ciência da Computação da Carnegie Mellon University, a Inteligência Artificial é a ciência e a engenharia de fazer computadores se comportarem de maneiras que, até pouco tempo atrás, nós pensávamos que era necessária inteligência humana (Forbes, 2021).

A respeito de *Machine Learning*, essa pode ser definida como sendo uma forma de estatística aplicada com maior ênfase no uso de computadores para estimar estaticamente funções complexas e uma ênfase menor em provar os intervalos de confiança em torno dessas funções (Goodfellow et al, 2016).

O *Deep Learning*, sendo uma abordagem do *Machine Learning*, utiliza uma hierarquia de conceitos que permite um computador aprender conceitos complexos a partir de outros mais simples. Se um gráfico fosse desenhado mostrando a ligação entre os conceitos construídos, teríamos um gráfico profundo e com muitas camadas (Goodfellow et al, 2016). A experiência que permite a construção de conceitos é o processamento de informações na etapa que é denominada como “treinamento”. Assim, o trabalho do ator humano é reduzido quando comparado ao *Machine Learning* tradicional.

2.2 Breve história do *Deep Learning*

Torsten Wiesel e David Hubel foram os ganhadores do prêmio Nobel em fisiologia no ano de 1981 devido às suas descobertas a respeito do processamento de informação no córtex visual em um trabalho realizado em 1979 (THE NOBEL PRIZE, 2021). Nesta importante descoberta, eles analisaram a atividade cerebral de um gato durante a apresentação de determinadas imagens simples e conseguiram captar com sucesso os impulsos que algumas imagens geraram nele. Esse avanço pode ser

considerado como a base para a ideia de composição de redes em camadas, de modo que as primeiras camadas fossem capazes de reconhecer padrões mais simples enquanto camadas mais profundas conseguissem reconhecer padrões mais complexos, utilizando-se da recombinação dos padrões mais simples reconhecidos pelas primeiras camadas.

Inspirado no trabalho de Wiesel e Hubel e depois em trabalhos próprios antecessores no mesmo tema, em 1980, Kunihiko Fukushima lança o Neocognitron (FUKUSHIMA, 1980), que consistia em uma rede neural artificial dividida em camadas com o objetivo do reconhecimento visual. A ideia de Fukushima foi de utilizar camadas de forma hierarquizada, seguindo muito do que foi apresentado por Wiesel e Hubel. Isso permitiu que o Neocognitron pudesse identificar caracteres escritos à mão.

Em 1998, Yann LeCun e Yoshua Bengio conseguiram mais um grande passo na história do *Deep Learning*. Com os insumos dos trabalhos supracitados somados a uma tecnologia mais avançada da época para processamento, uma maior base de dados e o algoritmo de propagação reversa (*Backpropagation*), não foi necessário nenhum código especializado para extração de características (*feature extraction)* e sim um maior foco na etapa de modelagem da rede. O modelo denominado de LeNet-5 desenvolvido por eles era capaz de aprender automaticamente os padrões das características (*features*) a partir dos dados fornecidos. Fazendo com que a etapa de engenharia de características (*feature engineering*) se tornasse obsoleta naquele contexto e rompendo assim com o *Machine Learning* tradicional que aplica mais tempo para a atividade de *feature engineering* e menos para a modelagem da rede.

Este marco foi alcançado na medida em que o algoritmo de *Backpropagation* (RUMELHART, HINTON e WILLIAMS, 1986) fez com que o rebalanceamento entre os elementos da rede neural artificial utilizada fosse possível a partir de cada iteração feita durante o treinamento da rede com a base de dados.

Entretanto, o ponto de virada principal foi em 2012, na terceira edição do Desafio de Reconhecimento Visual em Grande Escala da ImageNet (*ILSVRC - ImageNet Large Scale Visual Recognition Challenge*), quando a equipe de Geoffrey Hinton participou do desafio sendo o único com uma proposta baseada em *Deep Learning* enquanto todos os outros utilizaram modelos baseados em *Machine Learning* tradicional. O resultado foi expressivamente bem-sucedido, pois o modelo alcançou a menor taxa de erro entre os competidores e ajudou a colocar *Deep Learning* entre os assuntos mais relevantes de estudo na área de Inteligência Artificial.

Entre 2012 e a atualidade tivemos um grande desenvolvimento do *Deep Learning*. Este tipo de conhecimento tem sido valorizado por grandes empresas de redes sociais, e outras mais, que, por exemplo, a partir de sistemas de recomendação, buscam mostrar o anúncio mais compatível com o usuário. Enquanto isso, plataformas de *streaming* também usam sistemas de recomendação para manter seus usuários cada vez mais ativos. Os avanços não param por aí, a cada dia temos novos sistemas imitando ou superando

algumas de nossas capacidades humanas para nos oferecer um serviço ou produto que seja compatível com o nosso gosto a partir dos dados que esses sistemas adquiriram de nós mesmos. Além de sistemas de recomendação, o *Deep Learning* também é muito relevante sistemas de reconhecimento facial, de fala, de tradução entre línguas, etc.

2.3 Algumas Arquiteturas de Redes Neurais

2.3.1 Rede Neural Direta (*Feedforward*)

Uma rede neural artificial em sua forma mais direta e simples, ou *Feedforward* como é conhecida, será nosso primeiro exemplo de rede neural. Esta rede é dividida em três tipos de camadas. A camada de inserção de dados (*input layer*) é a camada inicial e o primeiro tipo. O segundo tipo de camada é a camada oculta (*hidden layer*), que pode ser uma única camada ou uma série de camadas, e o terceiro e último tipo de camada é a de apresentação do resultado (*output layer*).

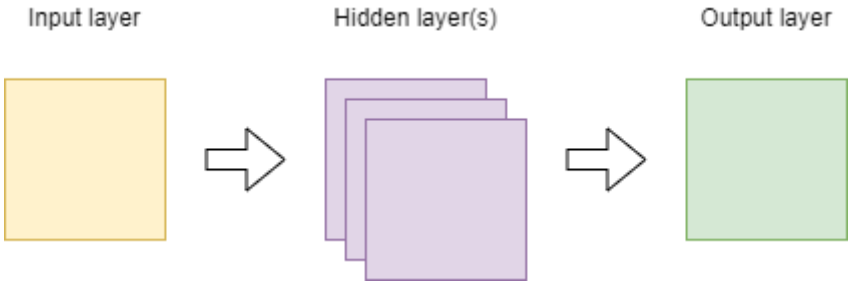

Figura 2 - Rede neural artificial (Feedforward).

Cada uma dessas camadas é formada por neurônios. Um neurônio é caracterizado por receber dados de outros neurônios, calcular a importância dos dados recebidos e propagar esse resultado para outros neurônios na rede. Para isso, eles são interligados e cada uma dessas ligações possui um peso que determina a relevância da ligação nos cálculos realizados. Dessa forma, ao criar uma rede neural nova, teremos os pesos desbalanceados e, durante a etapa de treinamento, a rede faz o balanceamento dos pesos. Em linhas gerais, é assim que se dá o processo de treinamento da rede neural.

Mais especificamente, no aprendizado supervisionado que é o utilizado neste trabalho, tal processo de aprendizagem acontece da seguinte forma: ao adicionar dados na *input layer, a* rede processa esses dados e apresenta um resultado na *output layer*. Cada rede utiliza uma função de perda (*loss function),* que indica o quão longe o resultado está do correto. Com esse valor, a rede aprende através do algoritmo de propagação reversa (*Backpropagation),* que faz o rebalanceamento das ligações entre os neurônios a fim de diminuir o valor do erro a cada iteração da etapa que é conhecida como treinamento.

De forma um pouco mais detalhada, o *Backpropagation* parte do gradiente da função de erro, tendo seu início na camada de saída e, de uma em uma, de trás para frente, os pesos são tem os seus valores reajustados de acordo com contribuição de cada peso para o erro, que é dado pelo gradiente desse peso em relação ao custo até a camada de entrada.

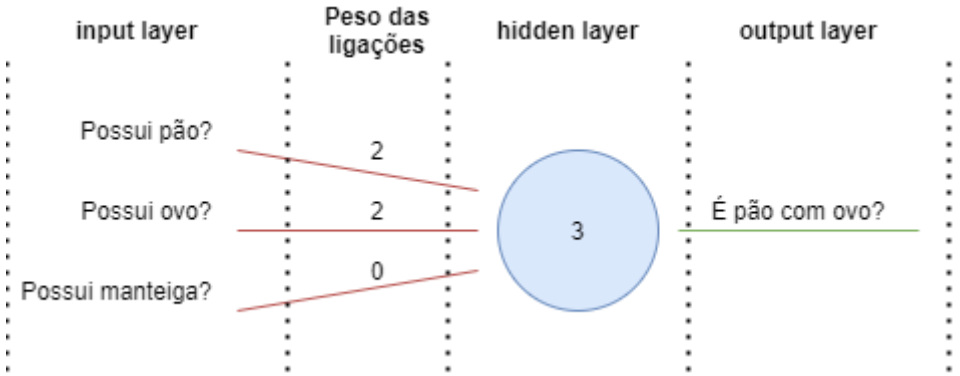

Figura 3 - Rede neural artificial capaz de identificar um pão com ovo.

Como exemplo, a Figura 3 mostra uma rede que é capaz de identificar um pão com ovo. Esta rede possui três neurônios na *input layer*, um na *hidden layer* e um na *output layer*. Os três neurônios da *input layer* respondem às seguintes perguntas com 'sim' ou 'não':

1. Possui pão?
2. Possui ovo?
3. Possui manteiga?

Cada pergunta com a resposta 'sim' ativará uma ligação entre a *input layer* e a *hidden layer*. Cada uma dessas ligações possui um peso que corresponde à sua importância para identificar se um item é um pão com ovo ou não, de modo que o peso dessas ligações é somado em cada neurônio da *hidden layer*, que neste caso possui apenas um neurônio. Caso essa soma atinja o valor mínimo estipulado pela *hidden layer* para identificar um pão com ovo, que no exemplo da Figura 3 é o valor 3, então esta envia um 'sim' para a *output layer* e vice-versa.

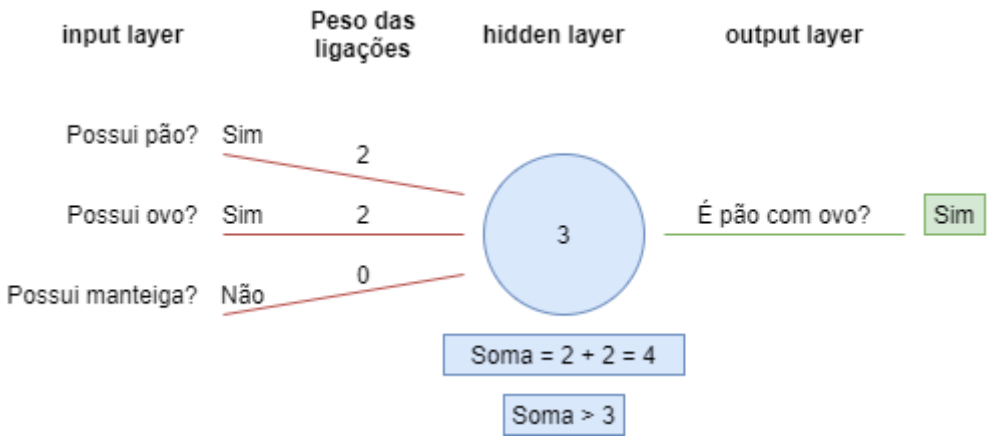

Figura 4 - Exemplo de funcionamento da rede neural artificial capaz de identificar um pão com ovo.

A Figura 4 mostra um caso em que a rede neural supracitada recebe dados na *input layer* que são identificados como um pão com ovo. Analisando essa rede, identificamos que possuir pão e possuir ovo possuem o mesmo peso: 2. Além disso, a única forma dessa rede neural apresentar um 'sim' na output layer é tendo um valor de soma dos pesos maior do que 3, o valor na *hidden layer*, e isso só é possível se a entrada fornecida possui pão e possui ovo.

De forma semelhante, podemos notar que possuir ou não manteiga é um dado irrelevante para essa rede neural na identificação de um pão com ovo. Isto é perceptível pois o peso da ligação que envolve possuir manteiga tem o valor igual a zero. Assim, independente do dado que tiver nesse neurônio da *input layer*, de nada vai agregar na soma desenvolvida no neurônio da *hidden layer*.

O exemplo mostrado nas Figuras 3 e 4 se trata de uma forma muito simplificada de como as redes neurais funcionam. Normalmente, as redes neurais reais possuem muitas outras especificidades que serão explicadas mais adiante neste trabalho.

2.3.2 Redes Neurais Recorrentes

2.3.2.1 Recurrent Neural Networks (RNN)

Uma rede neural recorrente ou *RNN* é um tipo de rede que tem como objetivo tratar cenários dotados de dados sequenciais. Dentre as aplicações para este tipo de rede, podemos citar algumas como: reconhecimento de texto através da fala, traduções de texto, previsão do tempo e predição de valores de mercado, assim como é usada neste trabalho.

Para entender como a *RNN* funciona, diga-se que exista um desafio para descobrir qual será a temperatura de amanhã no planeta Kepler-186f. Sem nenhuma informação a mais, esse desafio é dificílimo, porém caso as informações das temperaturas dos últimos dez dias estejam disponíveis, o desafio se torna bem mais factível.

Isso acontece porque a *RNN* possui um mecanismo de memória que permite à *hidden layer* armazenar os dados recebidos através da *input layer*, sendo que os dados recebem uma importância inversamente proporcional ao momento do tempo no qual foram recebidos pela rede. Assim, quanto mais antigo o dado recebido, menos importante ele é. O que possibilita essa memória é que, diferente de uma rede *feedforward*, essa utiliza um mecanismo de repetição que permite a reutilização do estado oculto (*hidden state*), que é a representação dos dados anteriormente processados (Phi, 2018a). A Figura 5 mostra a visão macro de uma *RNN* comparada a uma rede *feedforward.*

Devido a essas características, redes do tipo *RNN* "aprendem rapidamente, necessitando de poucas iterações de treinamento (menos de mil), além de precisarem de menos parâmetros e fornecerem melhor generalização." (TOSCANO e CABRAL, 2006).

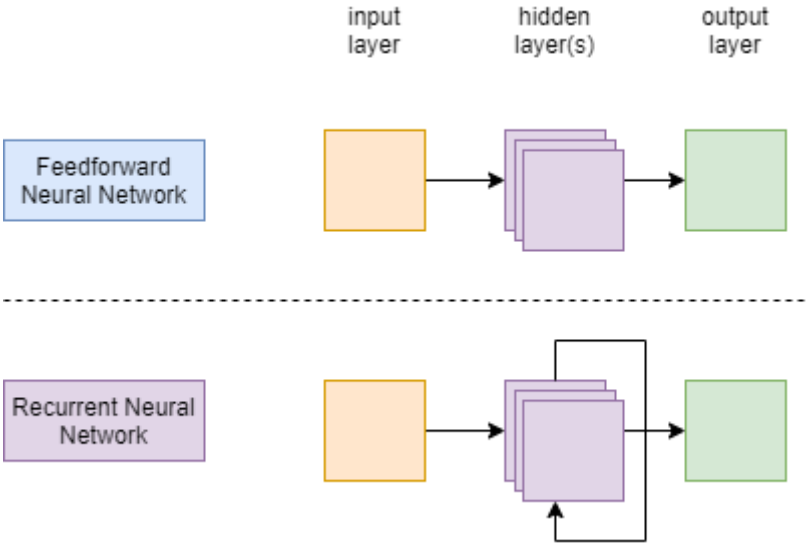

Figura 5 - Comparação macro da estrutura de uma rede *feedforward* e uma *RNN*.

Todavia, dois tipos de problema são muito comuns neste tipo de rede. São eles: o problema do gradiente explodindo (*Exploding Gradient*) e o problema do gradiente desaparecendo (*Vanishing Gradient*).

2.3.2.2 *Exploding Gradient* e *Vanishing Gradient*

Inicialmente, é necessário entender o que é o gradiente. O conceito do gradiente pode ser definido de mais de uma forma, porém uma simples e direta, utilizada por Donges (2021), é que: “um gradiente mede o quanto a saída de uma função muda se você alterar um pouco as entradas”.

Em uma rede neural ele é utilizado em conjunto com a função de perda (*loss function*) na etapa de treinamento. A *loss function* indica o quão distante o resultado determinado pela rede está do resultado esperado. O gradiente de erro indica a direção e a quantidade do ajuste numérico que deve ser feito nos pesos dos elementos da rede a fim de diminuir o valor indicado pela *loss function*.

Para Brownlee (2017), o problema do *exploding gradient* é quando, durante uma atualização, o gradiente de erro se acumula em um valor muito alto que gera grandes alterações nos pesos da rede e que gera um modelo instável por consequência. Nos melhores cenários, a rede deixa de conseguir aprender a partir dos dados de treinamento, todavia nos piores cenários os valores dos pesos podem ficar tão altos até o ponto de que ocasionarão um problema de estabilidade numérica, em que não serão mais reconhecidos pelo sistema computacional e não poderão mais ser atualizados

Por sua vez, o *vanishing gradient* é quando o gradiente assume um valor muito pequeno e irrelevante.

> Um gradiente pequeno significa que os pesos e as tendências das camadas iniciais não serão atualizados efetivamente com cada sessão de treinamento. Uma vez que essas camadas iniciais são frequentemente cruciais para reconhecer os elementos centrais dos dados de entrada, isso pode levar à imprecisão geral de toda a rede (WANG, 2019, tradução livre).[2]

Vale salientar que as camadas iniciais, que são as mais afastadas da camada de saída da rede, são as que sofrem menor influência do *Backpropagation* devido as suas contribuições para o resultado da rede serem as mais indiretas, já que são influenciadas por todas as outras camadas até a camada de saída da rede. Dessa forma, no problema do

[2] A small gradient means that the weights and biases of the initial layers will not be updated effectively with each training session. Since these initial layers are often crucial to recognizing the core elements of the input data, it can lead to overall inaccuracy of the whole network.

vanishing gradiente, o gradiente desses neurônios tende a zero e o balanceamento não ocorre para esses pesos.

A apresentação destes problemas é relevante para o presente trabalho na medida em que estes são recorrentes em *RNNs*, porém, pelo fato de não serem problemas novos, já existem muitos trabalhos e avanços que permitem contorná-los. A arquitetura do tipo Memória Curta de Longo Prazo (*LSTM - Long Short Term Memory*) é um ótimo exemplo disso. Essa rede é projetada para se livrar do problema de erro de desaparecimento (HOCHREITER; BENGIO, 2001, tradução livre).[3] Além da *LSTM*, tem-se a unidade recorrente fechada (*GRU - Gated Recurrent Unit)* como uma alternativa de arquitetura que também ataca os problemas supracitados.

2.3.2.3 Long Short-Term Memory (LSTM)

LSTM é um dos conceitos mais importantes deste trabalho. Esta seção é destinada a descrever essa arquitetura, qual problema ela visa a resolver e como isso é de fato resolvido. De forma geral, é apresentado todo o conhecimento a respeito dessa arquitetura que é necessário para o acompanhamento do restante deste trabalho.

Hochreiter e Schmidhuber (1997) introduziram o conceito de uma LSTM ao revisarem uma análise anterior realizada em 1991 do próprio Hochreiter que abordava o problema de *vanishing gradient* e suas implicações, tais como: a dificuldade que uma rede tinha para aprender e o tempo de treinamento que crescia consideravelmente em casos em que esse problema com respeito aos gradientes surgia.

A estrutura da *LSTM* foi melhorada com o passar dos anos. Atualmente, ela conta com um mecanismo de portas que possibilita a capacidade de redes aprenderem ultrapassando os problemas do *vanishing gradient* e do *exploding gradient* na maioria dos casos. Esse aprendizado ocorre com alguns passos e será abordado adiante de maneira simples. Um exemplo de uma rede neural do tipo *LSTM* é mostrado na Figura 6,

[3] LSTM is designed to get rid of the vanishing error problem.

esclarecendo que a multiplicação e a adição pontual significam multiplicação e a adição componente a componente respectivamente.

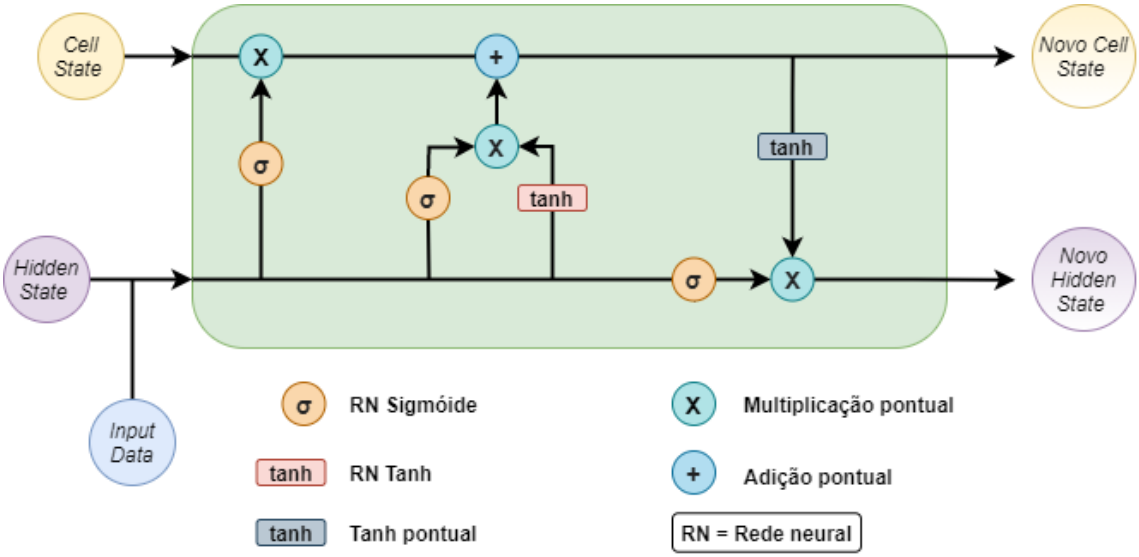

Figura 6 - Exemplo simplificado de uma rede neural do tipo LSTM.

O resultado de uma estrutura *LSTM* depende do processamento de três variáveis. São elas: o valor de entrada naquele passo do processamento (*input data)*, a memória de longo prazo atual da rede (*cell state*) e o resultado anterior naquele ponto da rede (*hidden state)*. O mecanismo de portas utilizado pelo processamento controla três portas: porta de entrada de dados (*input gate)*, porta de saída de dados (*output gate*), que é a responsável por mostrar o resultado, e a porta de esquecimento (*forget gate).*

O primeiro passo consiste em determinar se o *hidden state* combinado com a *input data* é algo irrelevante e deve ser descartado ou não para o *cell state*. Isso é feito com uma função de ativação sigmoide que resulta em um número entre 0 e 1, onde 0 significa que a informação é irrelevante e pode ser descartada e 1 significa o contrário. As funções utilizadas serão omitidas a fim de simplificar a explicação. Esse passo de poder descartar ou não esses dados é o mecanismo do já citado *forget gate* e pode ser visto na Figura 7.

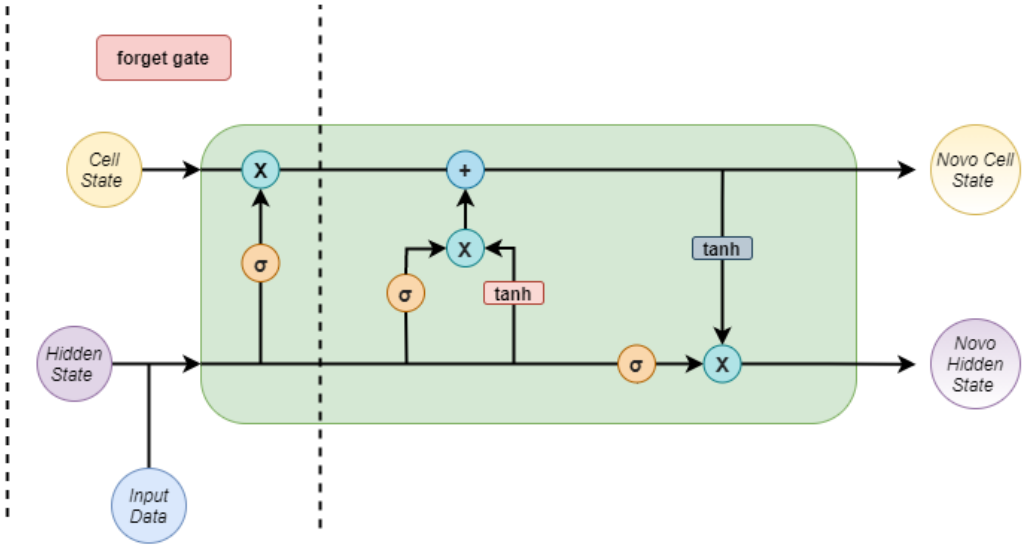

Figura 7 - Primeiro passo no processamento de uma LSTM.

O segundo passo tem como objetivo determinar qual será o novo *cell state* utilizando os mesmos dados de entrada que o *forget gate* utilizou, ou seja, o *hidden state* e o *input data*. Ele consiste em três etapas e é mostrado na Figura 8. Na primeira, uma função hiperbólica da tangente (*tanh)* é utilizada com o objetivo de criar uma nova memória para a *LSTM* e como seus dados de saída são entre -1 e 1, seguindo a mesma abordagem que na sigmoide, podemos aumentar ou diminuir a relevância de um "pedaço da memória". A segunda etapa é onde fica o *input gate* que com uma sigmoide determina quais partes do *input data* devem ser lembradas dado o *hidden state* atual. Já a terceira etapa é a combinação dos resultados da primeira e segunda etapa que resultam no novo *cell state*, que é a nova memória da *LSTM*.

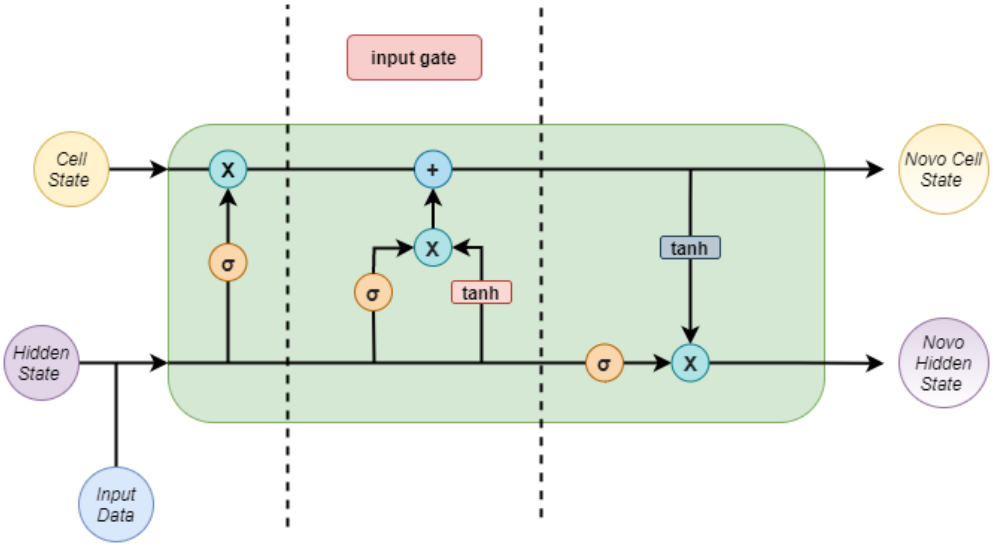

Figura 8 - Segundo passo no processamento de uma LSTM.

O terceiro e último passo atualiza o *hidden state* e é apresentado na Figura 9. Este novo valor é encontrado através da aplicação de *tanh* no *cell state*, que foi atualizado no passo anterior, combinado com o resultado de mais uma vez a aplicação da sigmoide no *input data* com o *hidden state* atual. Dolphin (2020) explica que para um *hidden state* na extremidade final de uma rede é necessário converter o seu valor em um resultado aceitável através de uma camada linear.

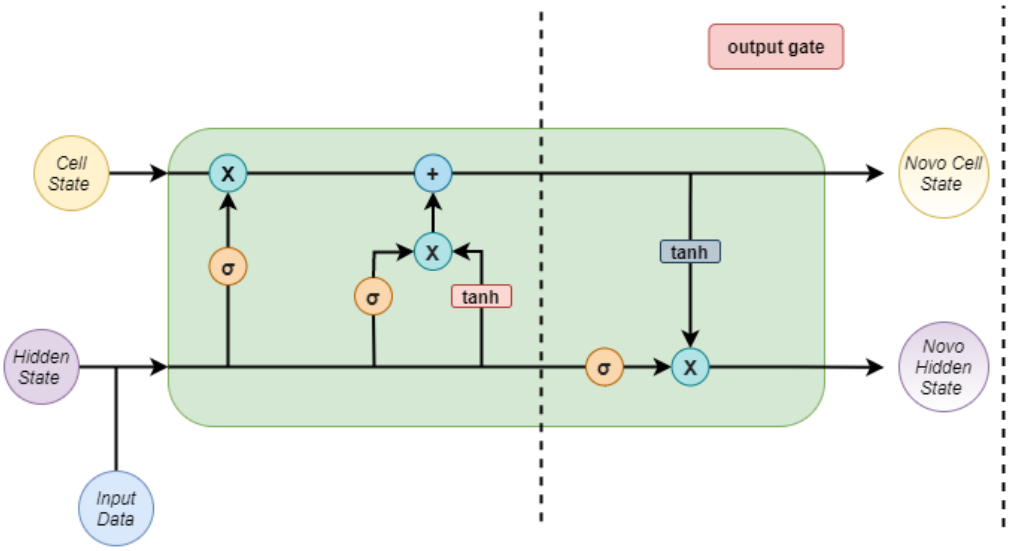

Figura 9 - Terceiro passo no processamento de uma LSTM.

Junior (2019), além de contribuir com o exemplo utilizado desta seção, pontua que existem variantes deste modelo mais simplista. “Uma popular variante LSTM, introduzida por Gers & Schmidhuber (2000), está adicionando “conexões peephole”.” Tais conexões permitem com que os *gates* possam utilizar o *cell state*. Outra variante da LSTM revisada por ele foi a GRU que será aprofundada a seguir.

2.3.2.4 Gated Recurrent Unit (GRU)

A *GRU* surgiu no trabalho de CHO et al (2014) e visa resolver o mesmo problema que a *LSTM*, o *vanishing gradient*. Sendo uma rede do tipo recorrente, é possível observar que alguns autores a reconhecem como um tipo de *LSTM* devido a semelhança em suas estruturas. Apesar da semelhança, pode-se dizer que uma *GRU* é mais simples que uma *LSTM* em sua estrutura simplificada e com um desempenho semelhante a esta, ou até melhor em alguns casos. Um exemplo simplificado de uma rede neural do tipo *GRU* pode ser visto na Figura 10, ressaltando que a multiplicação e a adição pontual significam multiplicação e a adição componente a componente respectivamente.

Diferentemente da *LSTM*, a *GRU* só possui duas portas. São elas: porta de redefinição (*reset gate*) e a porta de atualização (*update gate*). Apesar que a *GRU* também utiliza um mecanismo de portas semelhante. Além disso, não existe *cell state* nesta estrutura e o *hidden state* assume a responsabilidade pela informação em trânsito. As variáveis utilizadas são também são o *hidden state* anterior e o *input data.*

PHI (2018b), avalia o *update gate* como tendo o papel de atuar como o *forget gate* e o *input gate* da *LSTM* de uma só vez, já que ele é o responsável por determinar o que será adicionado no *hidden state* e o que será descartado, enquanto o *reset gate* é um mecanismo responsável por filtrar dados irrelevantes naquele ponto do processamento.

O processamento em uma *GRU* se inicia no *reset gate*, onde o *hidden state* anterior e o *input data* são processados por uma sigmoide que definirá o grau de importância de cada uma daquelas informações entre 0 e 1 as armazenando no *hidden state*. Em seguida, o *update gate* através da sigmoide também processa os mesmos dados e os soma de acordo com sua relevância ao *hidden state*, que também recebe o produto da aplicação de *tanh* aos dados iniciais que se soma ao valor final do *hidden state* neste ponto.

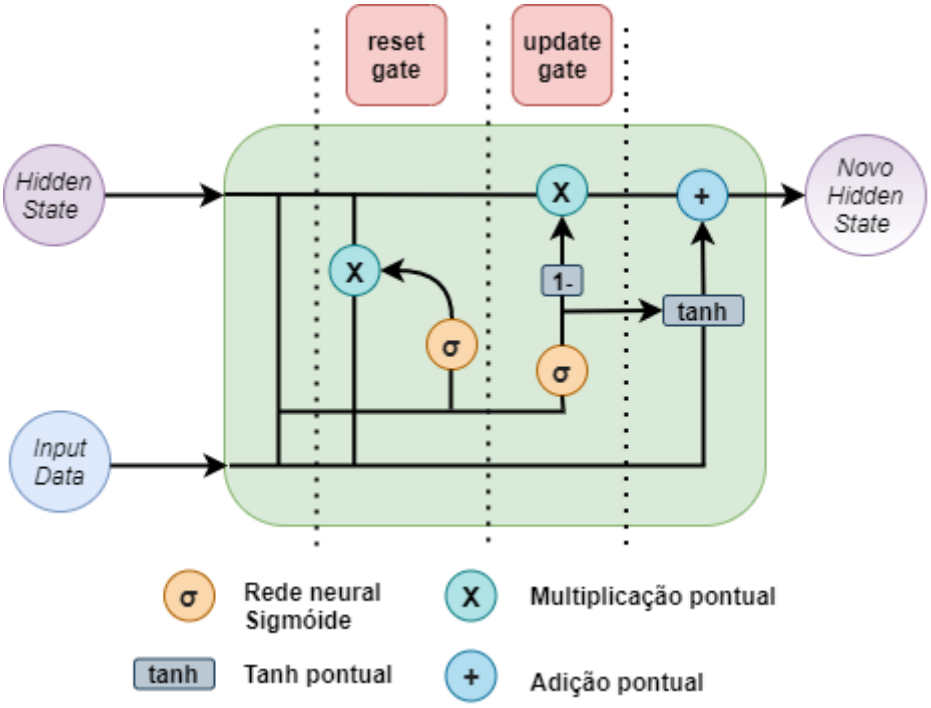

Figura 10 - Exemplo simplificado de uma rede neural do tipo GRU.

2.4 Mercado de Ações

"Uma ação representa a menor parcela do capital de uma empresa." (INFOMONEY 2, 2021). Quem adquire ações de uma empresa se torna sócio dela. Isso não quer dizer que aquele que as adquire tem automaticamente o poder de influenciar a empresa de forma significativa, até porque existem muitos sócios minoritários nas condições de pessoa física. Dessa forma, normalmente esse papel de influência fica para outras empresas no papel de investidores que possuem uma quantidade de ações muito relevante.

Entretanto, vale ressaltar que um acionista, até mesmo na condição de pessoa física e com uma participação percentual baixa no número de ações, adquire também direitos e deveres quando compra uma ação. O mesmo pode ser convidado para participar de consultas formais, nas quais os acionistas possuem um voto acerca de alguma questão com o peso similar à quantidade de ações que têm da empresa.

Acrescenta-se também o direito aos proventos financeiros da empresa, um grande atrativo para os investidores. Normalmente, empresas negociadas na bolsa de valores distribuem uma parte do lucro líquido para seus acionistas em forma de dividendos. O valor total a ser distribuído é dividido pelo número de ações daquela empresa e então cada acionista recebe de acordo com a porcentagem de ações detidas. Existem também outras formas de distribuição de proventos como juros sobre o capital próprio, bonificações e outras.

Além dos direitos, quando um acionista compra uma ação, ele está investindo o valor de compra da ação naquela empresa e a partir de então o preço da ação comprada pode valorizar ou desvalorizar, influenciando assim no preço de uma possível venda da ação por parte do acionista. Isso faz com que as ações sejam caracterizadas como produtos de renda variável, já que seu preço varia de acordo com o mercado ao longo do tempo.

Existem diferentes tipos de ações. O Infomoney lista mais de dez tipos (INFOMONEY 2, 2021), todavia as ações possuem características próprias que as

definem. Quando analisamos uma ação a partir de um determinado período de tempo, temos o que é normalmente chamado no mercado financeiro de indicadores. Entre eles podemos citar:

- **Nome da ação:** é representado por um código que no Brasil costuma ser quatro letras seguido de um número. Exemplos: CIEL3 (Cielo S.A.) e PETR4 (Petrobras - Petróleo Brasileiro S.A.).
- **Valor de abertura:** é o primeiro preço que a ação foi negociada, comprada por alguém e vendida por outra.
- **Valor de fechamento:** é o oposto do valor de abertura. É o último preço negociado.
- **Valor Máximo:** é o preço mais alto pelo qual aquela ação foi negociada.
- **Valor Mínimo:** é o menor preço pelo qual aquela ação foi negociada.
- **Volume:** é o número total de negociações daquela ação.

O presente trabalho apresenta como técnicas de *deep learning* podem ser aplicadas ao problema de previsibilidade da variação do preço de ações ao longo do tempo utilizando as informações provisionadas pelos seus indicadores. Abordagens variadas foram utilizadas e analisadas de acordo com seu desempenho.

3 **Revisão Bibliográfica**

Gavriel (2021) utilizou variações de arquiteturas de redes neurais do tipo *Long Short-Term Memory* (LSTM) para predizer valores do índice S&P 500, o índice que acompanha os 500 principais ativos negociados nas bolsas de valores americanas.

Uma das arquiteturas dentre as utilizadas foi a de camadas *LSTM* seguidas por camadas *dropout*, que realizam o desligamento de alguns neurônios da camada para cada rodada de treinamento, de modo a estimular um melhor aprendizado. Nesta arquitetura, a utilização de somente uma *feature*, classificação de informação disponibilizada pelo conjunto de dados utilizado, se mostrou mais eficaz do que a utilização de múltiplas *features*. A *feature* utilizada foi a do valor negociado do índice no momento do fechamento da bolsa de valores diariamente.

Outro tópico abordado foi o experimento com algoritmos de otimização que apresentavam melhor performance com a utilização de camadas de *LSTM* com *dropout* combinadas. Os algoritmos testados foram: Adadelta, Adam, Adamax, RMSprop (root mean square - erro quadrático médio) e SGD (stochastic gradient descent - gradiente descendente estocástico). Sendo o algoritmo Adam aquele que apresentou resultados melhores.

Saboor (2020) demonstrou em seu trabalho que técnicas de *Machine Learning* utilizando *LST*M apresentam melhores resultados do que métodos estatísticos convencionais como o modelo autorregressivo integrado de médias móveis (*ARIMA - Autoregressive integrated moving average*) para predizer valores de ações.

Nele foram utilizados os dados diários dos valores de ações de duas empresas diferentes que são negociadas em bolsas de valores diferentes, são elas: Honda Motor Co Ltd, negociada na bolsa de valores de Tóquio e Lenovo Group Limited, negociada na bolsa de valores de Hong Kong. Os dados totalizaram um pouco mais de dez anos de registros para ambas as ações e foram divididos em 75% dos dados para treinamento do modelo e 25% para teste do mesmo.

Diferentemente de Gavriel, Saboor não utilizou camadas *LSTM* combinadas com as camadas *dropout* e sim um modelo simples com duas camadas *LSTM* seguidas por uma *Dense*. Outra diferença foi a escolha do algoritmo de otimização Nadam, que é um algoritmo semelhante ao Adam, mas que usa outra abordagem de impulso. Ademais, o erro foi analisado sobre três perspectivas: Erro Médio Absoluto (MAE - *Mean Absolute Error*), Erro Médio Quadrático (*MSE - Mean Squared Error*) e pela raiz quadrada do Erro Médio Quadrático (*RMSE - Root Mean Squared Error*).

Em um trabalho de 2018, Shen et al. compararam o desempenho de quatro modelos diferentes e utilizaram três índices de bolsas de valores distintas para isso. Os índices utilizados foram: o S&P 500, o índice Hang Seng (*HSI - Hang Seng Index*), que é o principal índice da bolsa de Hong Kong (Suno, 2019), e o índice de ações alemão (*DAX - Deutscher Aktienindex*), que acompanha as 30 maiores empresas negociadas na bolsa de valores de Frankfurt. Dessa maneira, foi capaz de conseguir uma representação das Américas, da Europa e da Ásia.

Entre os modelos utilizados, dois se basearam em *GRU*, sendo que um deles foi uma combinação de *GRU* com Máquina de Vetores de Suporte (*SVM - Support Vector Machine*). Dos dois modelos restantes, um foi baseado em *SVM* e o último uma rede neural direta (*feedforward neural network*).

Após o treinamento dos modelos, foi visto que o modelo que combinou *GRU* com *SVM* foi o que melhor performou nos índices S&P 500 e HSI, enquanto o modelo baseado em *GRU* teve destaque com o índice *DAX*. Também é válido ressaltar que a rede neural direta se saiu melhor que o *SVM* em todos os testes. Esses testes nos mostraram resultados positivos para o uso de *GRU* para a tentativa de predizer valores de mercado, porém não foram feitas comparações com *LSTM*.

Sethia e Raut (2019) fazem essa comparação. Esse trabalho compara o desempenho de quatro modelos que se utilizam respectivamente de: *LSTM*, *GRU*, *SVM* e uma rede neural *feedforward* do tipo perceptron multicamadas (*MLP - Multi-Layer Perceptron*). O erro foi medido através do *MSE* demonstrado por cada modelo e os dados foram normalizados entre um intervalo de 0 até 1.

Os resultados apresentados mostram uma melhor performance do modelo proposto que utilizou *LSTM*, seguido do modelo que utilizou *GRU* e o modelo que utilizou *MLP* teve a pior performance. Algo interessante foi que os modelos com *LSTM* e *GRU* tiveram um resultado muito próximo, deixando assim mais espaço para que possa ser explorado nessa linha.

Song (2018) fez uma comparação próxima ao que Sethia e Raut fizeram, porém com algumas diferenças. Song utilizou dados de 2010 até 2017 de vinte diferentes empresas que têm suas ações negociadas em uma das duas bolsas de valores mais conhecidas. São elas a Associação Nacional de Corretores de Títulos de Cotações Automáticas (*NASDAQ - National Association of Securities Dealers Automated Quotations*) e a bolsa de valores de Nova Iorque (*NYSE - New York Stock Exchange*)

Esses dados foram normalizados entre o intervalo de -1 até 1. Além disso, oitenta por cento da base de dados foi destinada para a etapa de treinamento e os vinte por cento restante para a etapa de teste. Os modelos propostos para a comparação realizada

utilizaram: *LSTM, GRU, SVM* e aumento de gradiente extremo (*XGBoost - Extreme Gradient Boosting*).

Nos modelos que utilizaram *LSTM* e *GRU* foram utilizados *RMSE* para calcular o erro e Adam como algoritmo de otimização. Os modelos contaram com uma *input layer* seguida de duas *hidden layers* e uma *output layer* no final e utilizaram cinquenta épocas para o treinamento com o *batch size* igual a oitenta.

Para o modelo que utilizou *SVM*, Song utilizou a função linear como função do núcleo (*kernel*), já que a mesma apresentou uma taxa de erro menor que as demais e o parâmetro de regularização foi de 100, escolhido a fim de reduzir a chance de ter um problema de sobreajuste (*overfitting*) nos resultados. Já para o modelo com XGBoost foram utilizados os parâmetros: *gbtree* para o tipo de reforço de árvore (*tree booster*), o número de rodadas igual a mil, taxa de aprendizagem de 0.3 e regularização de 5. Os demais parâmetros foram utilizados com seus valores padrões.

Os resultados do trabalho de Song nos mostram que a taxa de acurácia média entre os modelos que utilizaram *LSTM* e *GRU* foram os que apresentaram melhores resultados compilando os resultados para as 20 ações utilizadas, sendo o modelo com *GRU* o melhor. O modelo com *XGBoost* foi o terceiro que obteve o melhor desempenho, porém com um resultado bem próximo ao modelo com *SVM*.

4 **Modelos Adotados**

Neste trabalho foram usadas cinco estruturas de modelos diferentes para serem realizados uma série de experimentos em cada um deles. Quatro dos cinco modelos utilizam LSTM, enquanto o modelo restante usa GRU. Quanto a LSTM e GRU, já abordamos em seções anteriores, entretanto veremos dois tipos de camadas que ainda não foram cobertas. São as camadas de eliminação (*dropout layer*) e a camada densa (*dense layer*).

A *dropout layer* é responsável por ocultar, de forma temporária, parte dos neurônios da *hidden layer* no processo de *Backpropagation* para ajudar na aprendizagem evitando o *overfitting*. Além disso, o *dropout* tem como função estimular a utilização de outros caminhos neurais, de modo que a rede não fique fortemente dependente de apenas um subconjunto de neurônios para realizar a sua lógica.

Por sua vez, a *dense layer* é uma simples camada conectada por completo (*fully connected*), na qual podemos indicar o número de saídas, o que faz o papel de traduzir todas as informações em um só valor quando a utilizamos como última camada do modelo.

As camadas de *dropout* foram utilizadas sempre em seguida a uma camada que seja LSTM ou GRU. Além disso, as camadas *dense* foram utilizadas no final de cada modelo podendo ser somente uma camada ou duas. A próxima imagem mostra os quatro modelos que utilizam LSTM.

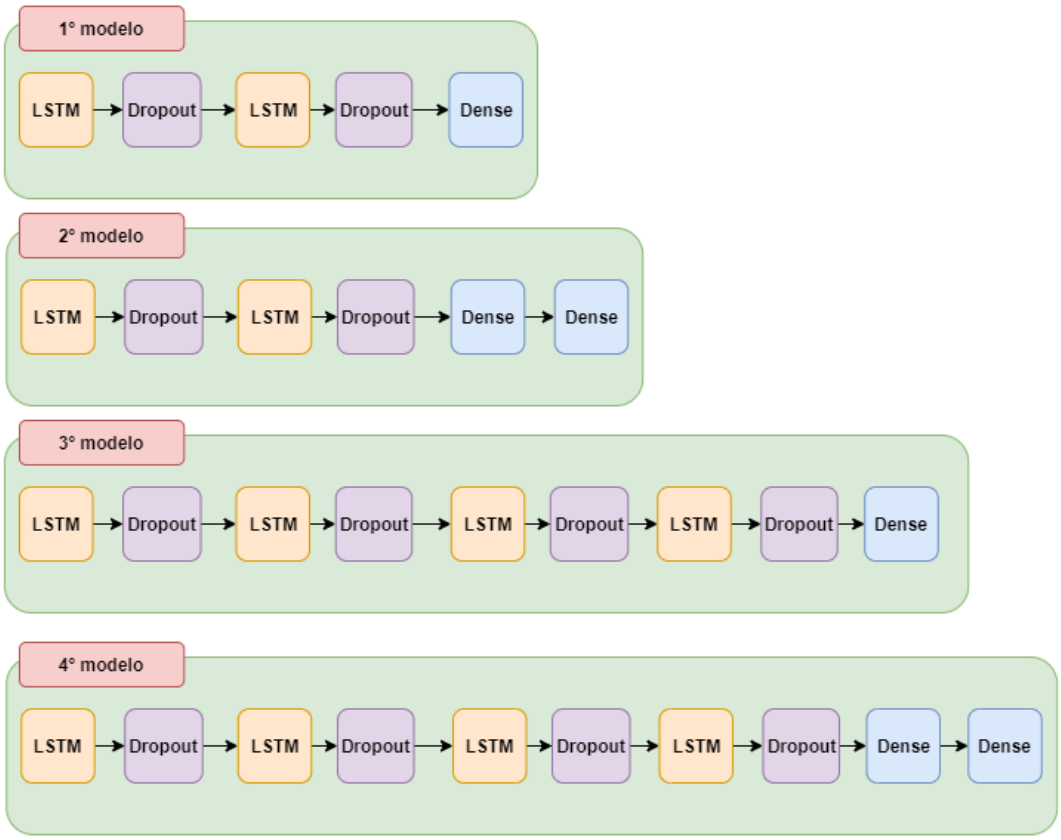

Figura 11 - Modelos que utilizam LSTM usados nos experimentos.

Temos assim o primeiro modelo com as camadas em sequência: LSTM, *dropout*, LSTM, *dropout* e *dense*. O segundo modelo conta com: LSTM, *dropout*, LSTM, *dropout*, *dense* e *dense*. O terceiro modelo contém: LSTM, *dropout*, LSTM, *dropout*, LSTM, *dropout*, LSTM, *dropout* e *dense*. Por fim, o quarto modelo possui: LSTM, *dropout*, LSTM, *dropout*, LSTM, *dropout*, LSTM, *dropout*, *dense* e *dense*. Todos os quatro modelos que utilizam LSTM são mostrados na Figura 11.

O terceiro e o quarto modelo foram criados com mais camadas propositalmente a fim de testar que modelos com um maior número de camadas leva potencialmente a um pior desempenho. Espera-se assim que os dois primeiros modelos alcancem melhores resultados que o terceiro e quarto modelo.

O quinto modelo é um modelo que se assemelha ao primeiro modelo quanto à estrutura de camadas, porém trocando as camadas LSTM por camadas GRU. Uma

ilustração deste modelo é encontrada na Figura 12. Essas duas estratégias foram adotadas para realizar comparações de desempenho.

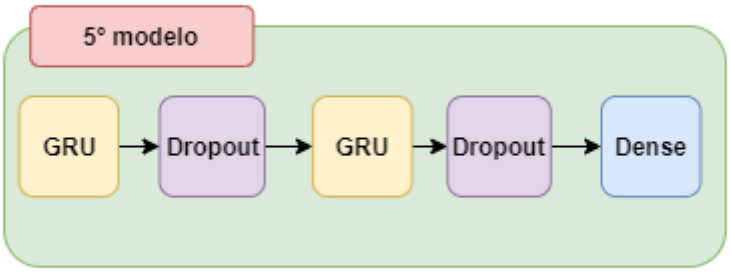

Figura 12 - Modelo usado que utiliza GRU nos experimentos.

5 Experimentos Computacionais

5.1 Dataset Utilizado

Para realizar os experimentos deste trabalho, foi utilizado o conjunto de dados (*dataset*), que está público, disponibilizado por Asutosh Nayak através da plataforma Kaggle (NAYAK, 2019).

Este *dataset* possui informações sobre as ações da empresa Microsoft Corporation, uma das maiores empresas de tecnologia do mundo. O nome da ação, ou seja, o código utilizado por esta é MSFT e a mesma é negociada na bolsa de valores americana *NASDAQ*.

Contendo as informações entre as datas de 02 de janeiro de 1998 até 02 de janeiro de 2019, o *dataset* disponibiliza todos os indicadores financeiros mencionados na Seção 3.1 sobre o mercado de ações. São eles: valor de abertura (*open*), valor de fechamento (*close*), valor máximo (*high*), valor mínimo (*low*), volume (*volume*), e por fim o dia que indica quando os demais indicadores ocorreram (*timestamp*).

Sendo assim, é totalizado um número de 5284 registros. Lembrando que as ações somente são negociadas nas bolsas de valores nos dias úteis, o que implica no fato de que o número de registros é significativamente inferior ao número de dias entre as datas de início e fim das informações disponibilizadas.

Os experimentos deste trabalho pedem a divisão do *dataset* em duas partes, uma disponibilizada para o treinamento do modelo e a segunda para os testes do modelo após o treinamento. Nesta divisão, foi utilizada a razão de 65% dos registros para treinamento do modelo e 35% para os testes.

Vale mencionar que metade dos experimentos foi realizada com o *dataset* completo, enquanto a outra metade utilizou somente 50% do *dataset*, mantendo a razão supracitada entre os dados de treinamento e de teste.

É importante ressaltar que neste trabalho foi priorizada a utilização de dados de alguma ação negociada na bolsa de valores brasileira, porém, devido a escassez de dados disponíveis com este perfil, foi escolhido seguir com dados da empresa Microsoft.

5.2 Ambiente Computacional

O ambiente computacional utilizado foi disponibilizado pelo Google através da plataforma "Google Colab"[4] que pode ser acessada gratuitamente. Esse serviço disponibiliza recursos computacionais incluindo unidades de processamento gráfico (*Graphics Processing Units* - GPUs).

O Google Colab é um serviço baseado no projeto de código aberto Jupyter e através dele é possível executar códigos na linguagem de programação Python em um navegador de internet. O navegador utilizado foi o Google Chrome, enquanto a versão da linguagem Python utilizada pela plataforma durante os experimentos foi a 3.6.9.

5.3 Códigos Utilizados e Detalhes dos Experimentos

Neste trabalho foram utilizados dois projetos durante a etapa dos experimentos. O primeiro projeto é uma aplicação criada com a linguagem de programação C#, e a ferramenta .Net Core, com o objetivo de gerar os identificadores dos experimentos. Este projeto foi necessário para otimizar o tempo de criação dos identificadores e evitar possíveis erros. O ambiente computacional utilizado para este projeto foi o sistema operacional Windows 10 Pro 64-bit, 8GB de memória RAM e processador Intel Core i7-7500U CPU.

No total, o número de identificadores foi de 480 e cada um dos identificadores é formado por uma junção de códigos conforme é mostrado no código da classe Program.cs que está disponível pela plataforma GitHub através do link: https://github.com/BrenoBaronte/RNNs-for-stock-market-prediction/blob/master/labels-generator/LabelsGenerator/Program.cs.

[4] Disponível em: <https://colab.research.google.com/>. Acesso em: 15 Nov. 2021

O código identificador de um experimento é formado por 7 partes e cada uma delas identifica um detalhe do experimento. O uso do código identificador nos experimentos foi importante para o agrupamento de determinados resultados na etapa de análise. A Tabela 1 apresenta cada uma das partes, mostrando a ordem desta no código, seu tamanho fixo, a descrição do código e os possíveis valores.

A parte “Tamanho dos datasets” é responsável por identificar quais *datasets* serão utilizados neste experimento. Quando for “T”, significa que os *datasets* utilizados têm o total dos dados, enquanto “H” significa que só possuem a metade deles.

“Quantidade de dias” é o valor que a rede neural leva em consideração quando for predizer um valor de *output*, consistindo no número de dias passados dos quais a rede utiliza os dados para predizer um novo valor.

“Algoritmo de otimização” e “Algoritmo de perda” são os algoritmos utilizados para otimização e perda no momento de compilação da rede neural. Para otimização, “A” representa o algoritmo Adam enquanto “S” representa o algoritmo do gradiente descendente estocástico. Para a perda, “MSE” representa o erro quadrático médio e “MAE” o erro médio absoluto.

“Épocas de treinamento” e “Tamanho do lote de treinamento” são os valores utilizados para o treinamento da rede neural criada. Para as épocas de treinamento, o “L” simboliza o 50, que está em algarismos romanos, e “C” o 100. Enquanto para o tamanho do lote de treinamento é o próprio valor que pode ser 16, 32 ou 64.

A parte de “Modelo de camadas” representa os modelos apresentados na seção anterior, sendo: “5C” - 1° modelo, “6C” - 2° modelo, “9C” - 3° modelo, “XC” - 4° modelo e “GC” o 5° modelo.

Ordem	Tamanho	Descrição do Código	Possíveis Valores
1°	1	Tamanho dos datasets	T, H
2°	2	Quantidade de dias	30, 60
3°	1	Algoritmo de otimização	A, S
4°	3	Algoritmo de perda	MSE, MAE
5°	1	Épocas de treinamento	L, C
6°	2	Tamanho do lote de treinamento	16, 32, 64
7°	2	Modelo de camadas	5C, 6C, 9C, XC, GC

Tabela 1 - Partes do código identificador gerado pelo primeiro projeto.

O segundo projeto, e o mais importante, é o código escrito na linguagem de programação Python que é capaz de decodificar o código identificador do primeiro projeto e então parametrizar e criar um modelo de rede neural de acordo com as informações passadas para ser treinado e avaliado.

Este código é utilizado no ambiente computacional descrito na seção 4.2 e está disponível pela plataforma GitHub através do link: https://github.com/BrenoBaronte/RNNs-for-stock-market-prediction/blob/master/RNNs-for-stock-prediction-notebook.ipynb.

O resultado de cada experimento é refletido em três variáveis de resultado. A primeira e mais importante para este trabalho é o erro do experimento, que é calculado utilizando o erro quadrático médio entre os valores previstos pela rede neural treinada e os valores reais do *dataset* de teste. Também medimos o tempo de treinamento da rede neural utilizada e o último valor de erro do treinamento. Os valores dos resultados dos experimentos estão disponíveis no Apêndice 1 e também é possível encontrá-los em: https://github.com/BrenoBaronte/RNNs-for-stock-market-prediction/blob/master/Results.csv.

5.4 Análise

A partir do resultado dos experimentos, que utilizaram todas as possíveis combinações diferentes com os parâmetros previamente apresentados entre os modelos, foi realizada uma análise a fim de identificar tendências e direcionamentos para se encontrar as combinações que possuem a menor taxa de erro na predição de valores de uma ação ao longo do tempo. Observa-se que quanto mais próximo de zero o valor do erro, melhor ele é.

Descrição	Valor	Identificador
Menor erro	1,130015334	T60AMAEL166C
Média dos erros	7,385837406	Não se aplica
Maior erro	32,1179887	H60AMAEC169C

Tabela 2 - Resultados absolutos dos experimentos.

A Tabela 2 mostra indicadores referentes aos erros dos experimentos. Assim, é possível identificar que o experimento que melhor performou utilizou os *datasets* com o total de registros, considerando 60 dias de informações antes de predizer o *output*, usando o algoritmo de otimização Adam, algoritmo de perda do erro médio absoluto, treinando 50 épocas com um tamanho de lote de 16 e utilizando o segundo modelo de camadas da rede neural.

O experimento que pior performou utilizou variáveis semelhantes ao experimento que obteve o melhor desempenho, diferenciando somente em: utilizar os *datasets* com a metade dos registros, ter um treinamento com 100 épocas e usar o terceiro modelo de camadas da rede neural. Isso mostra que determinadas variáveis possuem uma relevância muito alta no resultado final da performance do experimento. As Tabelas 3 e 4 mostram os 5 experimentos com os melhores resultados e os 5 com os piores resultados, respectivamente.

Descrição	Valor	Identificador
Menor erro	1,130015334	T60AMAEL166C
2° Menor erro	1,138433077	T60AMAEC646C
3° Menor erro	1,139430026	T30AMSEL166C
4° Menor erro	1,161324327	T30AMAEC645C
5° Menor erro	1,177804492	T30SMSEL165C

Tabela 3 - Os cinco experimentos com os melhores resultados (menores erros).

Descrição	Valor	Identificador
Maior erro	32,1179887	H60AMAEC169C
2° Maior erro	30,26264851	H30AMAEC169C
3° Maior erro	28,68150502	H60AMAEC16XC
4° Maior erro	25,68758751	H30AMAEC16XC
5° Maior erro	24,88089112	H60SMSEL649C

Tabela 4 - Os cinco experimentos com os piores resultados (maiores erros).

A partir desses valores é possível observar algumas tendências interessantes. Entre os piores resultados é notório a combinação de cinco das sete variáveis que quase não mudam. São elas “H”, “MAE”, “C”, “16”, combinados com “9C” ou “XC”. Significando que utilizar metade dos registros com o algoritmo de erro absoluto médio, 100 épocas de treinamento com o tamanho de lote de 16 no terceiro ou quarto modelo de camadas pode gerar um resultado consideravelmente negativo independentemente da quantidade de dias considerados no cálculo do *output* e do algoritmo de otimização.

Sobre o algoritmo de otimização, vale ressaltar que o algoritmo Adam apareceu em 4 dos 5 registros tanto na Tabela 3 como na Tabela 4, levando a resultados inconclusivos sobre seu uso nesses cenários em específico. Outra tendência notada, dessa vez na Tabela 3, é a combinação de “T” com a dupla “C” e “64” ou “L” e “16”, com o final “5C” ou “6C” produziram resultados satisfatórios. Isso significa a combinação do uso dos *datasets* com todos os registros, com a dupla de variáveis de treinamento 100 épocas com tamanho de lote de 64 ou 50 épocas com tamanho de lote de 16, em conjunto com o primeiro ou o segundo modelo de camadas.

Outro detalhe não muito surpreendente, porém muito importante, é a influência da quantidade de registros utilizados no treinamento para a performance dos modelos de redes neurais dos experimentos. O prefixo “T” em todos os identificadores da Tabela 3 e o prefixo “H” em todos os identificadores da Tabela 4 são as únicas constantes nesta amostra dos resultados que indicam que quanto mais dados são utilizados, melhor a rede neural pode aprender.

As Tabela 5 e 6 reforçam essa observação, mostrando que o maior erro encontrado na amostra de experimentos utilizando os *datasets* com todos os registros ainda é menor do que a média dos erros da amostra de experimentos que utilizaram os *datasets* com a metade dos registros.

Descrição	Valor	Identificador
Maior erro com dataset total	7,274095413	T30SMAEL646C
Média dos erros com dataset total	2,281188386	Não se aplica
Menor erro com dataset total	1,130015334	T60AMAEL166C

Tabela 5 - Dados dos erros dos experimentos que utilizaram os datasets com todos os registros.

Descrição	Valor	Identificador
Maior erro com metade do dataset	32,1179887	H60AMAEC169C
Média dos erros com metade do dataset	12,49048643	Não se aplica
Menor erro com metade do dataset	3,539484427	H30AMAEC646C

Tabela 6 - Dados dos erros dos experimentos que utilizaram os datasets com metade dos registros.

Adicionalmente, foi realizada uma análise a respeito dos modelos de camadas utilizados neste trabalho. Para isso, os experimentos foram divididos em grupos em que cada grupo possuía 5 elementos. Entre esses elementos, os identificadores têm um prefixo que só difere de grupo para grupo. Este prefixo tem tamanho 10 e contém desde as informações dos *datasets* até as informações do tamanho do lote utilizado no treinamento, deixando assim o modelo de camadas sendo o sufixo variável entre os elementos do grupo. A Figura 13 mostra um exemplo de grupo.

Analisando a performance por grupo, foram obtidos resultados inesperados. A Tabela 7 apresenta os resultados da soma do número de vezes em que cada modelo de camadas apresentou o melhor desempenho em cada um dos grupos dos experimentos. O 5° modelo, o único modelo de camadas que utilizou GRU no lugar de LSTM, foi o que teve os melhores resultados neste tipo de análise. Por outro lado, é intrigante que mesmo com essa superioridade, ainda assim não foi o suficiente para estar entre os 5 experimentos que obtiveram os menores erros conforme mostra a Tabela 3.

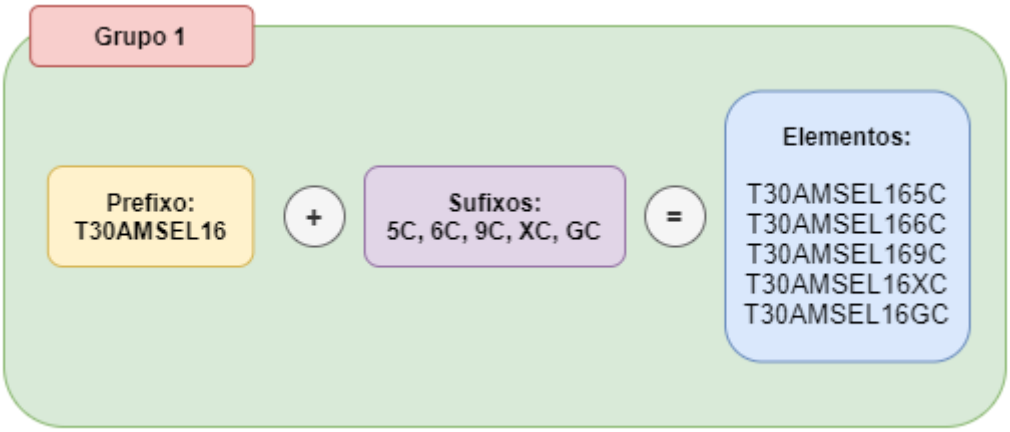

Figura 13 - Exemplo de grupo na análise dos modelos de camadas.

Modelo (Código)	Nº de vezes em que teve melhor performance
1° (5C)	13
2° (6C)	23
3° (9C)	3
4° (XC)	2
5° (GC)	55

Tabela 7 - Resultados do número de vezes em que cada modelo de camadas apresentou melhor desempenho.

6 Conclusão

6.1 Considerações finais

Foi possível implementar soluções para predizer os valores de uma ação ao longo do tempo com um grau de confiabilidade razoável, ainda que seja reforçado que não se pratiquem operações no mercado financeiro com os modelos de redes neurais apresentados neste trabalho.

A Tabela 3 mostra os principais modelos que obtiveram os melhores resultados e os três primeiros modelos tiveram uma performance próxima. É sugerido que, caso esses modelos venham a ser utilizados em outros trabalhos, se possível, realizem-se mais testes com os mesmos a fim de validar seu desempenho para cada cenário e não assumir que o que performou melhor nesses experimentos irá se repetir com outras bases de dados.

Desse modo, pudemos concluir inicialmente que o primeiro e o segundo modelos apresentados na Seção 4.3 foram os que tiveram os melhores resultados em performance, mas também que em uma segunda análise os experimentos com o quinto modelo que utilizou GRU no lugar de LSTM teve resultados surpreendentes e inesperados em comparação com os demais.

6.2 Limitações do projeto

Os modelos apresentados tiveram sua performance medida através do treinamento com somente um *dataset*, sendo em sua versão com metade ou a totalidade dos dados. Uma forma adicional e interessante de mensurar os modelos propostos seria treiná-los e testá-los com diferentes *datasets* a fim de investigar possíveis adaptações de diferentes modelos com outros dados.

Dentre os modelos propostos, foram apresentados 4 que utilizam LSTM e um que utiliza GRU. Sendo que o quinto modelo, que é o que usa GRU, possui a mesma estrutura

que o primeiro modelo, que utiliza LSTM, fazendo com que possuamos um paralelo entre eles. Uma das limitações deste trabalho foi a não extensão desse paralelo na investigação dos desempenhos entre LSTM e GRU para o segundo, terceiro e quarto modelo.

A extensão dos modelos, e consequentemente dos experimentos, seria muito interessante para analisar dados como os dispostos na Tabela 7. A criação de um modelo que utiliza GRU no lugar de LSTM, em uma estrutura semelhante ao do segundo modelo, seria um ótimo candidato a alcançar ótimos resultados, se baseando nos números que temos com os presentes experimentos.

Existem também outros tipos de redes neurais recorrentes conhecidas que trabalham com problemas de predição de valores. LSTMs do tipo bidirecional seriam uma boa adição aos experimentos realizados neste trabalho, por exemplo.

Por fim, a maior limitação deste trabalho foi a necessidade de interação humana para o começo de cada experimento na utilização do Google Collaboratory como ambiente computacional. Isso fez com que o número de experimentos não fosse maior, o que seria muito bem-vindo para as análises realizadas e para a disponibilização dos resultados.

6.3 Trabalhos futuros

Conforme mencionado acima, uma grande adição a este trabalho seria a realização de experimentos com um modelo semelhante ao segundo modelo proposto na Seção 4.3 que utilizasse GRU no lugar de LSTM.

Além disso, os dados disponibilizados pelo Apêndice 1 deixam em aberto inúmeras oportunidades de análises que não foram feitas neste trabalho. Outra proposta interessante é a tentativa de aperfeiçoamento dos modelos aqui propostos, uma vez que seus resultados não indicam que foi encontrado um novo padrão a ser seguido na elaboração de modelos que utilizam LSTM ou GRU.

A utilização dos modelos propostos em conjunto com o uso de *datasets* de outras empresas além da Microsoft seria uma extensão interessante deste trabalho a fim de qualificar que tipo de modelo, entre os disponibilizados, é o melhor para a tarefa de

predição de valores de maneira geral. Esta extensão também ajudaria a avaliar a questão do fator humano envolvido neste trabalho que não deve ser descartada.

Referências Bibliográficas

1. B3. b3.com.br. **Histórico pessoas físicas.** Disponível em: <http://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/consultas/mercado-a-vista/historico-pessoas-fisicas//> Acesso em: 2 Jan. 2021.

2. BROWNLEE, Jason. machinelearningmastery.com. **A Gentle Introduction to Exploding Gradients in Neural Networks.** Disponível em: <https://machinelearningmastery.com/exploding-gradients-in-neural-networks/> Acesso em: 26 Ago. 2021.

3. CHO, Kyunghyun; BENGIO, Yoshua; BOUGARES, Fethi; SCHWENK, Holger; BAHDANAU, Dzmitry. **Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation.** Disponível em: <https://arxiv.org/pdf/1406.1078v3.pdf> Acesso em: 17 Out. 2021.

4. DOLPHIN, Rian. towardsdatascience.com. **LSTM Networks | A Detailed Explanation.** Disponível em: <https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9> Acesso em: 11 Out. 2021.

5. DONGES, Niklas. builtin.com. **A Guide to RNN: Understanding Recurrent Neural Networks and LSTM Networks**. Disponível em: <https://builtin.com/data-science/recurrent-neural-networks-and-lstm> Acesso em: 26 Ago. 2021.

6. FORBES. Forbes.com. **Carnegie Mellon Dean Of Computer Science On The Future Of AI.** Disponível em: <https://www.forbes.com/sites/peterhigh/2017/10/30/carnegie-mellon-dean-of-computer-science-on-the-future-of-ai/?sh=c53c22821979> Acesso em: 2 Jan. 2021.

7. FUKUSHIMA, K. **Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.** Biol. Cybernetics 36, 193–202 (1980). https://doi.org/10.1007/BF00344251

8. GAVRIEL, Stylianos (2021) **Stock Market Prediction using Long Short-Term Memory.**

9. GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning.** Cambridge: MIT Press, 2016.

10. HOCHREITER, S.; BENGIO, Y. **Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies.** (2001).

11. HOCHREITER, Sepp & SCHMIDHUBER, Jürgen. (1997). **Long Short-term Memory. Neural computation.** 9. 1735-80. 10.1162/neco.1997.9.8.1735.

12. INFOMONEY. Infomoney.com.br. **Como funciona a bolsa de valores.** Disponível em: <https://www.infomoney.com.br/guias/como-funciona-a-bolsa-de-valores/> Acesso em: 2 Jan. 2021.

13. INFOMONEY 2. Infomoney.com.br. **Entenda como funciona o mercado de ações e a bolsa de valores.** Disponível em: <https://www.infomoney.com.br/guias/mercado-de-acoes/> Acesso em: 2 Jan. 2021.

14. INFOMONEY 3. Infomoney.com.br. **Renda Variável: um guia completo para conhecer e investir.** Disponível em: <https://www.infomoney.com.br/guias/renda-variavel/> Acesso em: 2 Jan. 2021.

15. JUNIOR, Jose R F. medium.com. **Redes Neurais Recorrentes — LSTM.** Disponível em: <https://medium.com/@web2ajax/redes-neurais-recorrentes-lstm-b90b720dc3f6> Acesso em: 11 Out. 2021.

16. MEDIUM. Medium.com. **2012: A Breakthrough Year for Deep Learning.** Disponível em: <https://medium.com/limitlessai/2012-a-breakthrough-year-for-deep-learning-2a31a6796e73> Acesso em: 2 Jan. 2021.

17. NAYAK, Asutosh. **Microsoft stock price (daily) 1998-2018.** Versão 1, 03 de Janeiro, 2019. Disponível em: <https://www.kaggle.com/darkknight91/microsoft-stock-price-daily-19982018> Acesso em: 22 Janeiro. 2021.

18. PHI, Michael. Towards Data Science. **Illustrated Guide to Recurrent Neural Networks.** Disponível em: <https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9> Acesso em: 21 Jul. 2021a.

19. PHI, Michael. Towards Data Science. **Illustrated Guide to LSTM's and GRU's: A step by step explanation.** Disponível em: <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21> Acesso em: 18 Out. 2021b.

20. PRESS RELEASE. NobelPrize.org. **Nobel Media AB 2021.** Disponível em: <https://www.nobelprize.org/prizes/medicine/1981/press-release/> Acesso em: 2 Jan. 2021.

21. RUMELHART, David & HINTON, Geoffrey & WILLIAMS, Ronald. (1986). **Learning representations by back-propagating errors.** Nature vol 323, 9 October 1986

22. SABOOR, Khalid and SABOOR, Qurat Ul Ain and HAN, Liyan and ZAHID, Abdul Saboor, **Predicting the Stock Market using Machine Learning: Long short-term Memory (2020).** Electronic Research Journal of Engineering, Computer and Applied Sciences, 2, 2020, 202-219, Available at SSRN: https://ssrn.com/abstract=3810128

23. SETHIA, Akhil & RAUT, Purva. (2019). **Application of LSTM, GRU and ICA for Stock Price Prediction: Proceedings of ICTIS 2018, Volume 2.** 10.1007/978-981-13-1747-7_46.

24. SHEN, Guizhu & TAN, Qingping & HAOYU, Zhang & ZENG, Ping & XU, Jianjun. (2018). **Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions.** Procedia Computer Science. 131. 895-903. 10.1016/j.procs.2018.04.298.

25. SONG, Y. (2018). **Stock Trend Prediction: Based on Machine Learning Methods.** UCLA. ProQuest ID: Song_ucla_0031N_16607. Merritt ID: ark:/13030/m5zs7sqk. Retrieved from https://escholarship.org/uc/item/0cp1x8th

26. SUNO. Suno.com.br. **Hang Seng: saiba mais sobre o principal índice da Bolsa de Hong Kong.** Disponível em: <https://www.suno.com.br/artigos/hang-seng/> Acesso em: 9 Oct. 2021.

27. TOSCANO, W.; CABRAL, E. L. L. **Contextualização de redes neurais recorrentes.** Exacta, São Paulo, v. 4, n. especial, p. 65-66, 25 nov. 2006.

28. WANG, Chi-Feng. towardsdatascience.com. **The Vanishing Gradient Problem.** Disponível em: <https://towardsdatascience.com/the-vanishing-gradient-problem-69bf08b15484> Acesso em: 26 Ago. 2021.

Apêndice 1

Número do Experimento	Código	Tempo de treinamento (s)	Erro de teste	Último erro de treinamento
1	T30AMSEL165C	309,2992144	1,267476537	0,0005662778276
2	T30AMSEL166C	335,7674155	1,139430026	0,0005809877184
3	T30AMSEL169C	648,0317595	1,269935849	0,0007074894384
4	T30AMSEL16XC	671,3946266	1,281312926	0,0007808553055
5	T30AMSEL16GC	341,6910245	1,652206666	0,0005691424594
6	T30AMSEL325C	175,6805449	1,46467644	0,0005697978777
7	T30AMSEL326C	153,1958838	1,510104573	0,0005930616753
8	T30AMSEL329C	334,1662521	2,14005614	0,0006706135464
9	T30AMSEL32XC	350,5328231	1,298944847	0,0007203138666
10	T30AMSEL32GC	222,9520109	1,483519802	0,0005974632222
11	T30AMSEL645C	99,55468893	1,675306439	0,0005682730116
12	T30AMSEL646C	100,3081293	1,282643084	0,0006594962324
13	T30AMSEL649C	227,3261757	2,969784336	0,0008066240116
14	T30AMSEL64XC	236,5578325	2,634929372	0,0007435891312
15	T30AMSEL64GC	101,1345243	1,225892401	0,0006457719719
16	T30AMSEC165C	502,0826375	1,498728868	0,0005688433885
17	T30AMSEC166C	568,6335263	1,437130018	0,000517124834
18	T30AMSEC169C	1207,482234	2,089870015	0,0005901011755
19	T30AMSEC16XC	988,5425799	1,638430786	0,0006963916821
20	T30AMSEC16GC	571,0674117	1,466594543	0,0005104313605
21	T30AMSEC325C	323,4639959	1,798525436	0,0005698634777
22	T30AMSEC326C	332,1223183	1,360894365	0,0005476305378
23	T30AMSEC329C	614,5424821	1,657811704	0,0006078952574
24	T30AMSEC32XC	669,5287495	1,585794122	0,000636310433
25	T30AMSEC32GC	332,239984	1,576113567	0,0005414522602
26	T30AMSEC645C	214,4431465	1,231710368	0,0005187013885
27	T30AMSEC646C	222,2375641	1,767071929	0,0006150329718
28	T30AMSEC649C	461,8396604	2,119162569	0,0006122276536
29	T30AMSEC64XC	451,8678751	1,886658603	0,0006424030871
30	T30AMSEC64GC	195,8387034	1,288610791	0,0005548264598
31	T30AMAEL165C	286,8333244	1,18598433	0,011960214
32	T30AMAEL166C	276,4175408	1,384410677	0,01266172901
33	T30AMAEL169C	588,0238612	2,575964877	0,01396618225
34	T30AMAEL16XC	597,0752623	1,298976696	0,01411629375

35	T30AMAEL16GC	270,4083302	1,281568049	0,01211935375
36	T30AMAEL325C	157,2052822	1,338855994	0,01275309268
37	T30AMAEL326C	165,4505708	1,767479235	0,01202071179
38	T30AMAEL329C	345,7158892	2,262167251	0,01350567397
39	T30AMAEL32XC	364,285213	1,805476602	0,01327804942
40	T30AMAEL32GC	221,014045	1,211759739	0,0121538816
41	T30AMAEL645C	107,9967782	1,306675183	0,01186076179
42	T30AMAEL646C	110,3768532	1,980015743	0,01184697356
43	T30AMAEL649C	251,3930683	1,510545596	0,01335844956
44	T30AMAEL64XC	228,0556679	2,183291876	0,01481145434
45	T30AMAEL64GC	90,99966526	1,282597822	0,01186560094
46	T30AMAEC165C	531,4470136	1,418166075	0,01123648044
47	T30AMAEC166C	525,0616684	2,135409506	0,01200497989
48	T30AMAEC169C	1147,57757	1,526840564	0,0124907475
49	T30AMAEC16XC	1242,252998	1,554819432	0,0128689073
50	T30AMAEC16GC	565,1676259	1,626965224	0,01121715643
51	T30AMAEC325C	333,3457589	1,241770565	0,01144506689
52	T30AMAEC326C	301,8564181	1,226119336	0,01167389192
53	T30AMAEC329C	709,1865659	2,697896336	0,01173110958
54	T30AMAEC32XC	624,7024195	1,239302673	0,0125201242
55	T30AMAEC32GC	325,3002043	2,115170944	0,01125402097
56	T30AMAEC645C	210,6775949	1,161324327	0,01141125988
57	T30AMAEC646C	196,0203724	2,668874911	0,01114400383
58	T30AMAEC649C	389,0398068	2,281373643	0,01281893346
59	T30AMAEC64XC	461,1182177	1,39298469	0,01242798008
60	T30AMAEC64GC	179,5055728	1,278949259	0,01137321442
61	T30SMSEL165C	206,9233055	1,177804492	0,01241029892
62	T30SMSEL166C	273,2668695	2,230511498	0,001622164971
63	T30SMSEL169C	457,0880218	1,284125868	0,01199201588
64	T30SMSEL16XC	558,445209	3,252120772	0,00264236005
65	T30SMSEL16GC	292,8899143	1,644931882	0,001007069135
66	T30SMSEL325C	221,3304782	2,448737396	0,001804494881
67	T30SMSEL326C	221,7955155	2,557024561	0,002016221872
68	T30SMSEL329C	327,9617898	3,543946939	0,002813965315
69	T30SMSEL32XC	388,5818043	3,529335506	0,002933969256
70	T30SMSEL32GC	205,5902047	1,765002402	0,001070829108
71	T30SMSEL645C	107,9942994	2,505627652	0,00194168021
72	T30SMSEL646C	101,7336152	2,635960698	0,001875515678
73	T30SMSEL649C	227,5152354	4,167326551	0,003167062532
74	T30SMSEL64XC	221,9069326	3,2081947	0,003061177675
75	T30SMSEL64GC	91,23671627	1,543795271	0,00122983777
76	T30SMSEC165C	545,9855089	2,979327714	0,001428386779
77	T30SMSEC166C	625,4533944	2,36289375	0,001411988866

78	T30SMSEC169C	1189,18534	2,953217441	0,002491380088
79	T30SMSEC16XC	1111,381915	3,589774087	0,002379050246
80	T30SMSEC16GC	641,9932795	1,524106385	0,0008935671649
81	T30SMSEC325C	325,3415444	2,254303113	0,001661148388
82	T30SMSEC326C	329,9177113	2,81813194	0,001633076463
83	T30SMSEC329C	705,1897209	3,613458363	0,002737099072
84	T30SMSEC32XC	704,2300849	3,667466557	0,002790911123
85	T30SMSEC32GC	402,2568245	1,577591015	0,00104091817
86	T30SMSEC645C	206,7031963	2,4194276	0,001843587495
87	T30SMSEC646C	199,8235793	2,405887664	0,001691311831
88	T30SMSEC649C	403,8974562	3,737204604	0,003187543945
89	T30SMSEC64XC	446,437973	3,147016273	0,002972349292
90	T30SMSEC64GC	283,9690421	1,661269702	0,001211357303
91	T30SMAEL165C	325,4114449	2,567784275	0,0186416544
92	T30SMAEL166C	284,9491642	2,187060422	0,02184066921
93	T30SMAEL169C	500,0384505	3,391834122	0,02493658476
94	T30SMAEL16XC	548,700026	2,730163649	0,02540506423
95	T30SMAEL16GC	308,2043562	1,64403835	0,01801183075
96	T30SMAEL325C	205,5352719	2,176015814	0,01929317042
97	T30SMAEL326C	167,0435801	2,999500017	0,02206922509
98	T30SMAEL329C	327,1293328	3,024117489	0,02686885372
99	T30SMAEL32XC	309,1918926	3,939758315	0,02706276625
100	T30SMAEL32GC	152,3211961	4,225653096	0,01873525232
101	T30SMAEL645C	93,28620458	3,26369202	0,02342105657
102	T30SMAEL646C	109,7817636	7,274095413	0,03101078793
103	T30SMAEL649C	210,1886075	5,466759301	0,03039530106
104	T30SMAEL64XC	204,0150816	5,194639219	0,03421098366
105	T30SMAEL64GC	114,922044	1,50554729	0,0259511359
106	T30SMAEC165C	499,2216563	2,488296107	0,01673196442
107	T30SMAEC166C	554,4476676	3,035507596	0,01673962921
108	T30SMAEC169C	1289,122133	2,349778949	0,02159786969
109	T30SMAEC16XC	1203,019623	3,422010987	0,02264144085
110	T30SMAEC16GC	604,061311	1,830468924	0,01577196829
111	T30SMAEC325C	325,7841072	2,259795643	0,01727205329
112	T30SMAEC326C	325,0852084	2,991869332	0,01898490265
113	T30SMAEC329C	630,8044159	2,450070981	0,02459499612
114	T30SMAEC32XC	701,8209174	6,140926483	0,02573035099
115	T30SMAEC32GC	322,0815272	2,66102102	0,01827913523
116	T30SMAEC645C	221,5975482	3,620634399	0,01943445578
117	T30SMAEC646C	199,2154791	3,727224359	0,0215111468
118	T30SMAEC649C	464,9462719	4,584607372	0,02768107504
119	T30SMAEC64XC	447,7285433	4,808257644	0,02939804085
120	T30SMAEC64GC	202,1468072	1,626149168	0,02003063075

121	T60AMSEL165C	567,1983938	2,237773262	0,0005819168873
122	T60AMSEL166C	558,8638952	1,842962142	0,0006125313812
123	T60AMSEL169C	988,2136419	2,193160074	0,0007636856753
124	T60AMSEL16XC	1108,503824	1,822280299	0,0007598812226
125	T60AMSEL16GC	459,069608	1,209248797	0,0005969743943
126	T60AMSEL325C	325,449245	1,787764842	0,0005813722964
127	T60AMSEL326C	310,42222	1,206723849	0,0006411049399
128	T60AMSEL329C	628,6015728	1,887501336	0,0007296050899
129	T60AMSEL32XC	594,3898494	2,318673002	0,0008287405944
130	T60AMSEL32GC	338,9296939	1,610463937	0,0005568641354
131	T60AMSEL645C	205,4497268	1,521515881	0,0005953903892
132	T60AMSEL646C	197,4193897	1,636935954	0,0005915683578
133	T60AMSEL649C	406,6676924	1,6291626	0,0007056691684
134	T60AMSEL64XC	449,1429772	1,949094535	0,0007993364707
135	T60AMSEL64GC	191,9777343	1,570773017	0,0005833503092
136	T60AMSEC165C	919,341558	1,291731936	0,0005208867951
137	T60AMSEC166C	1125,639493	1,66104028	0,0005805341643
138	T60AMSEC169C	2250,401752	1,543776587	0,0005801821244
139	T60AMSEC16XC	2345,205195	2,954914887	0,0006606571842
140	T60AMSEC16GC	1093,073991	1,577449076	0,0005253125564
141	T60AMSEC325C	686,1020823	1,374423203	0,0005436770734
142	T60AMSEC326C	602,2416463	1,742466482	0,0005652162363
143	T60AMSEC329C	1349,500399	1,21667037	0,0006270923768
144	T60AMSEC32XC	1288,816411	1,278251648	0,0006568725221
145	T60AMSEC32GC	667,9072409	1,753280192	0,000490312581
146	T60AMSEC645C	463,7457495	1,419941463	0,000585414411
147	T60AMSEC646C	401,6627131	1,796356353	0,000567601528
148	T60AMSEC649C	824,3775954	1,340794341	0,0007023286307
149	T60AMSEC64XC	820,0861068	1,262355211	0,0007032712456
150	T60AMSEC64GC	360,4782438	1,279207124	0,0005126948236
151	T60AMAEL165C	609,8488529	1,927995535	0,01188777853
152	T60AMAEL166C	626,0613184	1,130015334	0,01277778577
153	T60AMAEL169C	1168,628768	1,390361884	0,01418711897
154	T60AMAEL16XC	1020,362916	1,310340514	0,01469401736
155	T60AMAEL16GC	594,1116519	1,279634892	0,01158328447
156	T60AMAEL325C	302,7248988	1,312063664	0,01206299756
157	T60AMAEL326C	386,2539237	2,529733471	0,01249742042
158	T60AMAEL329C	670,8532741	2,044851296	0,01322611235
159	T60AMAEL32XC	688,7567637	1,558867234	0,0154795507
160	T60AMAEL32GC	326,2331905	1,406478898	0,01172049809
161	T60AMAEL645C	187,1601083	1,313766532	0,01265950408
162	T60AMAEL646C	205,9813662	1,26368522	0,01216700021
163	T60AMAEL649C	449,4917059	1,510670789	0,013567416

164	T60AMAEL64XC	448,5031908	2,002661138	0,01392393652
165	T60AMAEL64GC	183,0336864	1,286421591	0,01090042759
166	T60AMAEC165C	1019,584763	1,361044876	0,01193166338
167	T60AMAEC166C	1045,398433	1,337000722	0,01132356282
168	T60AMAEC169C	2144,951693	1,572353157	0,01268948708
169	T60AMAEC16XC	2254,701468	1,505932184	0,01377825253
170	T60AMAEC16GC	1164,298696	1,316728526	0,01117522549
171	T60AMAEC325C	746,7888324	1,472821617	0,01151689328
172	T60AMAEC326C	748,2249384	1,479892418	0,01197194774
173	T60AMAEC329C	1235,765335	1,753590945	0,01219806448
174	T60AMAEC32XC	1250,112494	1,596338772	0,01287867408
175	T60AMAEC32GC	685,6791751	1,223470488	0,011234615
176	T60AMAEC645C	445,7211242	1,602209252	0,01101039443
177	T60AMAEC646C	402,2156093	1,138433077	0,0108743282
178	T60AMAEC649C	792,5843351	1,636043261	0,01276420802
179	T60AMAEC64XC	761,7910252	2,043963985	0,01336904429
180	T60AMAEC64GC	385,483063	1,345354779	0,01082106493
181	T60SMSEL165C	480,9499383	2,621425119	0,001676636282
182	T60SMSEL166C	521,3822124	2,205912271	0,001643056283
183	T60SMSEL169C	1048,779982	3,362570561	0,002932787407
184	T60SMSEL16XC	1027,503702	4,291878571	0,002685815329
185	T60SMSEL16GC	625,5239711	1,541065262	0,001071712933
186	T60SMSEL325C	298,9145818	2,740314581	0,00186824007
187	T60SMSEL326C	295,6521595	2,424997653	0,001854183502
188	T60SMSEL329C	636,9912403	4,58238741	0,003121101065
189	T60SMSEL32XC	628,3470805	3,736146099	0,002888646908
190	T60SMSEL32GC	325,6744328	1,584421606	0,001238315017
191	T60SMSEL645C	205,186856	2,7304749	0,002103064675
192	T60SMSEL646C	179,3084495	2,764077797	0,00233356026
193	T60SMSEL649C	388,9877748	3,764011756	0,003384076757
194	T60SMSEL64XC	448,4861801	3,663628307	0,003246632637
195	T60SMSEL64GC	188,8107326	1,531166145	0,001331104431
196	T60SMSEC165C	1009,086613	2,234684849	0,001414924394
197	T60SMSEC166C	1002,076818	2,271401264	0,001302760793
198	T60SMSEC169C	2084,535094	3,572900669	0,002500261879
199	T60SMSEC16XC	2024,244033	2,996679744	0,002252844395
200	T60SMSEC16GC	1258,535555	1,525919266	0,0009569177055
201	T60SMSEC325C	609,9464571	2,320523446	0,001586862607
202	T60SMSEC326C	608,098115	2,330398643	0,001678098924
203	T60SMSEC329C	1184,576858	4,491096692	0,002908294089
204	T60SMSEC32XC	1304,023345	3,001227124	0,002633902011
205	T60SMSEC32GC	650,4581296	1,591503589	0,001123688649
206	T60SMSEC645C	386,3234367	2,674941183	0,001910039806

207	T60SMSEC646C	407,596236	2,392213674	0,001945845317
208	T60SMSEC649C	823,8742719	3,840424068	0,003062710864
209	T60SMSEC64XC	825,064455	3,375469893	0,002980428049
210	T60SMSEC64GC	476,9320235	1,559953154	0,00123384106
211	T60SMAEL165C	582,9279606	1,976099809	0,01876865141
212	T60SMAEL166C	520,8563781	1,912235787	0,01972597837
213	T60SMAEL169C	964,4020054	2,460066052	0,02426345274
214	T60SMAEL16XC	1147,925578	5,333753132	0,02614997327
215	T60SMAEL16GC	508,0961161	1,911755778	0,0180775784
216	T60SMAEL325C	316,269105	3,261662554	0,02001419105
217	T60SMAEL326C	320,2361476	4,155416336	0,0239575617
218	T60SMAEL329C	604,1588476	2,662991644	0,02724767663
219	T60SMAEL32XC	571,7796104	4,717125456	0,03062281571
220	T60SMAEL32GC	329,8290994	3,398325857	0,02032646164
221	T60SMAEL645C	197,1187537	2,836803189	0,02258444764
222	T60SMAEL646C	206,1359904	2,911857094	0,02578071877
223	T60SMAEL649C	448,8654754	2,79141831	0,03070600703
224	T60SMAEL64XC	382,201154	4,985459357	0,03238097206
225	T60SMAEL64GC	209,7595601	4,075299319	0,02295893058
226	T60SMAEC165C	920,9231005	2,350720706	0,01635411382
227	T60SMAEC166C	1165,192793	2,27990847	0,01782562025
228	T60SMAEC169C	2249,608492	2,330482888	0,02205342427
229	T60SMAEC16XC	2100,50977	5,481069399	0,02177001908
230	T60SMAEC16GC	987,2643447	2,020705943	0,01556782611
231	T60SMAEC325C	586,1865597	1,941605265	0,0173350051
232	T60SMAEC326C	601,35483	2,003958979	0,02043437399
233	T60SMAEC329C	1288,318845	3,30811601	0,02358577214
234	T60SMAEC32XC	1349,326853	4,130361525	0,02468049154
235	T60SMAEC32GC	627,7479649	2,629058764	0,01759378426
236	T60SMAEC645C	388,5194774	2,513682413	0,01905892417
237	T60SMAEC646C	372,1532717	2,686918242	0,02102507837
238	T60SMAEC649C	709,4887402	2,859955075	0,02833925188
239	T60SMAEC64XC	868,0713704	3,074680472	0,0296141915
240	T60SMAEC64GC	387,1341898	2,75662029	0,02020932734
241	H30AMSEL165C	153,6838744	6,112459584	0,0008247370715
242	H30AMSEL166C	141,9778154	4,318910069	0,0007349524531
243	H30AMSEL169C	266,6119823	17,32634731	0,001040780451
244	H30AMSEL16XC	281,2713089	10,21924081	0,001116383704
245	H30AMSEL16GC	168,2162502	5,470536968	0,000638621219
246	H30AMSEL325C	82,16801667	7,314680414	0,001088345307
247	H30AMSEL326C	83,93546128	5,753048273	0,0007648873725
248	H30AMSEL329C	175,7495301	14,60588484	0,001392439008
249	H30AMSEL32XC	209,1854541	9,439679276	0,001085589989

250	H30AMSEL32GC	95,8574388	6,20594274	0,0009266950656
251	H30AMSEL645C	50,27912068	7,922475543	0,001433585654
252	H30AMSEL646C	50,7002759	6,375229321	0,0009969198145
253	H30AMSEL649C	110,3023658	14,33981406	0,002036473015
254	H30AMSEL64XC	149,2687802	9,958797527	0,001254316652
255	H30AMSEL64GC	52,93234801	6,687597599	0,00150641182
256	H30AMSEC165C	280,2617705	7,532336711	0,0006636573817
257	H30AMSEC166C	277,0444539	5,701066839	0,0005917777307
258	H30AMSEC169C	628,1276605	22,08208682	0,0007671005442
259	H30AMSEC16XC	516,9734478	20,69749044	0,0007295481628
260	H30AMSEC16GC	306,0604928	5,31380565	0,0007074528839
261	H30AMSEC325C	166,5065711	4,937390397	0,0006318711094
262	H30AMSEC326C	220,7736304	3,816790003	0,0006127469824
263	H30AMSEC329C	343,1355066	14,65333887	0,0009432375082
264	H30AMSEC32XC	344,0124066	10,47642202	0,0009005719912
265	H30AMSEC32GC	207,5071669	5,306876307	0,000567865558
266	H30AMSEC645C	86,11442876	6,55112232	0,001041967655
267	H30AMSEC646C	103,0960712	5,067728089	0,0007504732348
268	H30AMSEC649C	211,7689896	14,0924896	0,001222279505
269	H30AMSEC64XC	198,4096842	9,250079312	0,0008621871821
270	H30AMSEC64GC	109,4857142	5,554090003	0,0009026640328
271	H30AMAEL165C	151,2372556	6,820276554	0,01792749017
272	H30AMAEL166C	125,1053944	3,845435123	0,01886444911
273	H30AMAEL169C	281,0895786	22,82195994	0,02247001417
274	H30AMAEL16XC	272,4062514	19,21155734	0,02186595462
275	H30AMAEL16GC	153,4051208	7,704149088	0,01800031029
276	H30AMAEL325C	73,25456262	7,495396124	0,02066905051
277	H30AMAEL326C	145,7704468	4,418746132	0,01911482215
278	H30AMAEL329C	149,5009079	13,8406549	0,02502809651
279	H30AMAEL32XC	162,9908805	13,22021492	0,02485167608
280	H30AMAEL32GC	77,61633992	5,71775468	0,01841228083
281	H30AMAEL645C	52,63504481	6,437223587	0,02428266592
282	H30AMAEL646C	50,78063393	5,068049501	0,01964342594
283	H30AMAEL649C	122,2118587	12,02257303	0,02906457894
284	H30AMAEL64XC	95,18849921	9,143931542	0,02467253432
285	H30AMAEL64GC	54,84774065	5,157149204	0,02200846002
286	H30AMAEC165C	279,0043175	9,45681127	0,01718703285
287	H30AMAEC166C	271,6289387	8,348562303	0,01778388955
288	H30AMAEC169C	568,5589402	30,26264851	0,01751696877
289	H30AMAEC16XC	689,6202524	25,68758751	0,01985472068
290	H30AMAEC16GC	276,5220888	6,942395644	0,01768441498
291	H30AMAEC325C	141,2149005	4,560459625	0,01760187745
292	H30AMAEC326C	158,3148048	6,924299626	0,01916205324

293	H30AMAEC329C	322,1264877	19,4975143	0,01833780296
294	H30AMAEC32XC	329,0693891	17,58330302	0,02008625492
295	H30AMAEC32GC	178,8329632	5,997056605	0,01710224152
296	H30AMAEC645C	145,976146	4,644215237	0,01849293523
297	H30AMAEC646C	87,8761847	3,539484427	0,01784431562
298	H30AMAEC649C	211,3949063	14,93546784	0,02106712386
299	H30AMAEC64XC	268,5878887	8,642019894	0,02019099705
300	H30AMAEC64GC	121,8523457	5,977971625	0,0170960594
301	H30SMSEL165C	128,1487472	17,43305962	0,002016282408
302	H30SMSEL166C	159,9780571	16,11142894	0,001642539515
303	H30SMSEL169C	284,7390807	20,0476588	0,002749580191
304	H30SMSEL16XC	345,0323067	18,85268841	0,002833240665
305	H30SMSEL16GC	132,5736511	6,959887595	0,001685475465
306	H30SMSEL325C	94,8000071	16,94141504	0,002313556615
307	H30SMSEL326C	163,0050349	17,72653756	0,002314922633
308	H30SMSEL329C	223,8493938	22,25743033	0,003292272333
309	H30SMSEL32XC	155,7338982	20,98476498	0,002844052622
310	H30SMSEL32GC	85,99408603	8,885867609	0,002128720982
311	H30SMSEL645C	53,86477137	19,34668412	0,002691267524
312	H30SMSEL646C	54,48941946	16,61856202	0,002118779346
313	H30SMSEL649C	118,556776	21,85649383	0,003617835697
314	H30SMSEL64XC	99,7907505	20,9198877	0,003260462545
315	H30SMSEL64GC	70,3262496	7,960751999	0,002504911041
316	H30SMSEC165C	282,9266458	14,78567176	0,001623110031
317	H30SMSEC166C	296,6448119	12,32360427	0,001397551387
318	H30SMSEC169C	523,7534873	20,87437217	0,002546216594
319	H30SMSEC16XC	480,2487669	17,86124934	0,002433716087
320	H30SMSEC16GC	314,236871	6,089457319	0,001228887006
321	H30SMSEC325C	223,117389	17,10984479	0,001968314406
322	H30SMSEC326C	154,7964787	13,38959491	0,001820374047
323	H30SMSEC329C	328,5273798	20,34580328	0,002894763136
324	H30SMSEC32XC	328,8255355	17,8183108	0,002379343845
325	H30SMSEC32GC	205,898128	6,412781764	0,00145808782
326	H30SMSEC645C	89,2701602	18,52500808	0,002170388587
327	H30SMSEC646C	106,8043191	17,52751003	0,001787150279
328	H30SMSEC649C	191,8551257	21,99148555	0,003237546887
329	H30SMSEC64XC	213,5136106	21,84270815	0,003088472178
330	H30SMSEC64GC	145,736237	7,693337124	0,002070863731
331	H30SMAEL165C	130,0129099	11,473821	0,02720482834
332	H30SMAEL166C	117,1777575	12,19411877	0,02685372159
333	H30SMAEL169C	305,9335222	19,82334298	0,035923969
334	H30SMAEL16XC	277,5633636	21,60513088	0,03554873541
335	H30SMAEL16GC	153,3157175	5,510374141	0,02407896519

336	H30SMAEL325C	85,57817221	12,34297266	0,0292397812
337	H30SMAEL326C	80,66605997	13,44617989	0,03112343699
338	H30SMAEL329C	146,5123944	20,94854789	0,03650137782
339	H30SMAEL32XC	174,0559084	22,44640958	0,03940042108
340	H30SMAEL32GC	148,2476859	6,374357547	0,02499178052
341	H30SMAEL645C	48,45077014	14,39636899	0,03059385344
342	H30SMAEL646C	54,23586464	13,83271087	0,0349948816
343	H30SMAEL649C	106,8808351	21,68215297	0,03945469111
344	H30SMAEL64XC	148,2030323	22,13602428	0,03906263039
345	H30SMAEL64GC	47,15417337	7,101288259	0,03024318442
346	H30SMAEC165C	326,445785	9,263609136	0,02425814606
347	H30SMAEC166C	291,5752087	10,41920331	0,02634521946
348	H30SMAEC169C	584,6258762	18,17007863	0,0310254693
349	H30SMAEC16XC	543,2086	20,30209794	0,03312402219
350	H30SMAEC16GC	325,9397454	5,316095488	0,02240323834
351	H30SMAEC325C	225,3009405	10,97177509	0,0268933624
352	H30SMAEC326C	164,5057986	11,08632474	0,02620118484
353	H30SMAEC329C	312,6347237	19,43000625	0,03350850195
354	H30SMAEC32XC	343,5466599	20,66005854	0,03480647132
355	H30SMAEC32GC	181,1977191	5,402808843	0,02432363853
356	H30SMAEC645C	99,40460515	12,77326902	0,02801108547
357	H30SMAEC646C	126,2870998	13,03878302	0,02987454832
358	H30SMAEC649C	230,8012445	21,77073131	0,0369695276
359	H30SMAEC64XC	187,2354438	20,74293138	0,03847335279
360	H30SMAEC64GC	104,8700984	5,581874867	0,02528537437
361	H60AMSEL165C	265,1001072	7,9938061	0,0008275432047
362	H60AMSEL166C	221,5633028	7,435330426	0,0006811183412
363	H60AMSEL169C	569,2240775	17,15274891	0,001029901905
364	H60AMSEL16XC	568,6675918	10,90420971	0,001027096063
365	H60AMSEL16GC	327,4205122	6,323278179	0,0006264464464
366	H60AMSEL325C	141,2415757	5,404743702	0,001121607726
367	H60AMSEL326C	142,7440281	5,08108228	0,0007441900671
368	H60AMSEL329C	270,7291675	14,00977666	0,00147126615
369	H60AMSEL32XC	329,4305608	8,81814194	0,001184654888
370	H60AMSEL32GC	205,3288004	4,316427199	0,0008375414764
371	H60AMSEL645C	145,4664285	7,079091419	0,001775249024
372	H60AMSEL646C	83,73693085	6,000672976	0,000885512738
373	H60AMSEL649C	178,0266659	12,10807571	0,001938051893
374	H60AMSEL64XC	163,2108555	11,26975841	0,001376903034
375	H60AMSEL64GC	124,4807467	7,055276982	0,001289092237
376	H60AMSEC165C	626,0352836	7,548813494	0,0006034901598
377	H60AMSEC166C	520,7929676	6,391648979	0,0005981111899
378	H60AMSEC169C	1038,365146	21,8141326	0,0007655560039

379	H60AMSEC16XC	920,2736249	24,5709766	0,0008503147983
380	H60AMSEC16GC	596,8047986	5,302361871	0,0005288737011
381	H60AMSEC325C	265,9152744	6,571827061	0,0006591417477
382	H60AMSEC326C	386,1965556	5,024839236	0,0006318400847
383	H60AMSEC329C	598,5657809	12,32885001	0,0008828920545
384	H60AMSEC32XC	628,8345072	8,938653318	0,0008590202197
385	H60AMSEC32GC	360,3797121	4,590225532	0,0005535228411
386	H60AMSEC645C	167,2609234	8,626543768	0,001156965503
387	H60AMSEC646C	158,2243907	5,105480493	0,0006903973408
388	H60AMSEC649C	451,5826402	11,83194192	0,001376389991
389	H60AMSEC64XC	448,7450168	6,986227374	0,0009650522261
390	H60AMSEC64GC	218,1372957	5,707467393	0,0008514297078
391	H60AMAEL165C	252,8824453	5,971107205	0,01933181658
392	H60AMAEL166C	220,8445809	4,774016558	0,01824509911
393	H60AMAEL169C	461,3801308	19,91413398	0,02042816021
394	H60AMAEL16XC	569,9633806	18,72121836	0,02164086141
395	H60AMAEL16GC	199,6783035	5,58589945	0,01815225929
396	H60AMAEL325C	205,3879898	6,520742419	0,02004770003
397	H60AMAEL326C	221,7500427	4,156167928	0,01889220625
398	H60AMAEL329C	302,2219648	13,57755726	0,02478767745
399	H60AMAEL32XC	344,6569793	12,69912272	0,02254955098
400	H60AMAEL32GC	206,6929564	5,014819798	0,01750908606
401	H60AMAEL645C	161,6482189	7,434506856	0,02492026053
402	H60AMAEL646C	161,1630673	5,148820749	0,01790818758
403	H60AMAEL649C	208,6697907	13,20082684	0,02811974287
404	H60AMAEL64XC	183,9315186	9,398655459	0,0230522044
405	H60AMAEL64GC	146,238251	5,133249478	0,0234064348
406	H60AMAEC165C	489,666153	10,79567776	0,0168561507
407	H60AMAEC166C	493,7491407	9,25830979	0,01739703864
408	H60AMAEC169C	897,5138214	32,1179887	0,01931946911
409	H60AMAEC16XC	1008,866378	28,68150502	0,01883307844
410	H60AMAEC16GC	627,3620474	6,525692496	0,01704964228
411	H60AMAEC325C	325,0884411	6,106835965	0,01659428887
412	H60AMAEC326C	326,0888486	4,909599269	0,01644108817
413	H60AMAEC329C	628,6894622	19,13575833	0,01866575703
414	H60AMAEC32XC	552,9374452	17,19259864	0,02068676427
415	H60AMAEC32GC	216,7863257	5,501091695	0,01657592878
416	H60AMAEC645C	205,8849516	6,301216471	0,0186811313
417	H60AMAEC646C	175,8954909	3,574757788	0,01896422915
418	H60AMAEC649C	388,4148662	10,5927058	0,02105495147
419	H60AMAEC64XC	381,5753636	8,497248541	0,02002448402
420	H60AMAEC64GC	206,316051	6,999879611	0,01634406485
421	H60SMSEL165C	221,3135524	14,47587066	0,001822447404

422	H60SMSEL166C	250,9667201	12,14960655	0,001677131397
423	H60SMSEL169C	548,7623141	20,35633545	0,002759467578
424	H60SMSEL16XC	522,4628203	19,45829431	0,002527413191
425	H60SMSEL16GC	326,7638683	7,098332307	0,001477304264
426	H60SMSEL325C	171,9651921	18,00635425	0,002144324826
427	H60SMSEL326C	223,2600183	15,36781282	0,001917552436
428	H60SMSEL329C	286,4164884	22,47253959	0,003093831474
429	H60SMSEL32XC	328,7991726	20,01363698	0,002656277036
430	H60SMSEL32GC	171,6092889	7,861265147	0,0020575854
431	H60SMSEL645C	96,66043591	22,55981064	0,002765695564
432	H60SMSEL646C	98,33091831	16,02567144	0,002113651251
433	H60SMSEL649C	269,6769435	24,88089112	0,003906974103
434	H60SMSEL64XC	174,86044	22,96645857	0,00312047638
435	H60SMSEL64GC	72,76957989	7,712408278	0,002328222152
436	H60SMSEC165C	565,4388878	16,07957504	0,001616265858
437	H60SMSEC166C	500,6062872	12,12812924	0,001428513206
438	H60SMSEC169C	1108,217923	17,69026218	0,002293348778
439	H60SMSEC16XC	1109,320023	16,63476954	0,002287490526
440	H60SMSEC16GC	626,9560075	6,268704289	0,001161238644
441	H60SMSEC325C	285,3815584	14,31897356	0,001865486265
442	H60SMSEC326C	326,4352965	13,71828314	0,001615779591
443	H60SMSEC329C	607,0919716	20,65805101	0,002699412405
444	H60SMSEC32XC	628,7492523	17,94524951	0,002236908069
445	H60SMSEC32GC	326,0140135	6,456871624	0,001542601851
446	H60SMSEC645C	205,9200671	18,45764742	0,002212856896
447	H60SMSEC646C	169,3353786	15,60750982	0,001991938567
448	H60SMSEC649C	449,076438	22,34857406	0,003125860589
449	H60SMSEC64XC	395,423696	21,30418697	0,00277641695
450	H60SMSEC64GC	206,3416274	7,689953307	0,001877589384
451	H60SMAEL165C	263,1752911	10,64869453	0,02650237456
452	H60SMAEL166C	290,1401336	11,88287013	0,02645548433
453	H60SMAEL169C	508,2509553	19,24478927	0,03484092653
454	H60SMAEL16XC	569,3679202	21,8676164	0,03607742488
455	H60SMAEL16GC	299,11637	6,590546581	0,02469230071
456	H60SMAEL325C	206,1850941	12,7250339	0,02721861936
457	H60SMAEL326C	156,8889618	11,30519274	0,03003544174
458	H60SMAEL329C	328,5308216	21,3630955	0,03718991205
459	H60SMAEL32XC	288,2947953	22,38775711	0,04079794511
460	H60SMAEL32GC	114,921382	7,435930347	0,02822224237
461	H60SMAEL645C	98,33349633	15,32062943	0,0337321423
462	H60SMAEL646C	97,65701485	16,58002265	0,0380740799
463	H60SMAEL649C	208,8352995	22,70042539	0,04112847522
464	H60SMAEL64XC	209,6300428	20,70552725	0,04208011553

465	H60SMAEL64GC	147,0430422	6,351440413	0,03433258832
466	H60SMAEC165C	490,3422425	8,662473904	0,02421878465
467	H60SMAEC166C	566,282954	9,300522778	0,02542866953
468	H60SMAEC169C	1031,167416	17,01296525	0,0314427726
469	H60SMAEC16XC	1108,512261	20,40072155	0,03246337175
470	H60SMAEC16GC	563,6356246	5,293485058	0,02152811177
471	H60SMAEC325C	325,966408	9,498548792	0,02519889362
472	H60SMAEC326C	325,7692983	11,96337394	0,02735851333
473	H60SMAEC329C	628,9882205	17,95711528	0,03371146321
474	H60SMAEC32XC	629,7685766	20,2022269	0,03707591817
475	H60SMAEC32GC	205,1516683	5,55494432	0,02372029983
476	H60SMAEC645C	171,4398367	12,20650415	0,02734751441
477	H60SMAEC646C	206,1502569	14,76677706	0,0285886731
478	H60SMAEC649C	388,3461556	19,24245608	0,03597841784
479	H60SMAEC64XC	448,7322593	20,89902802	0,03865126148
480	H60SMAEC64GC	266,4135456	6,45394458	0,02615732141